Programas de juegos con el ZX Spectrum y Spectrum+

W. Simister



EL ORDENADOR PERSONAL

ceac

Programas de juegos con el ZX Spectrum y Spectrum+



Programas de juegos con el ZX Spectrum y Spectrum+

W. Simister



Perú, 164 - 08020 Barcelona - España

Traducción autorizada de la obra:
HOW TO WRITE ZX SPECTRUM AND SPECTRUM + GAMES PROGRAMS

Editado en lengua inglesa por
Bernard Babani Ltd.

ISBN 0-85934-132-1

Perú, 164 - 08020 Barcelona (España)

Primera edición: Febrero 1987

ISBN 84-329-7018-2

Depósito Legal: B-4398 - 1987

Impreso por
GERSA, Industria Gráfica
Tambor del Bruc, 6
08970 Sant Joan Despí (Barcelona)

Printed in Spain
Impreso en España

INDICE

PROLOGO

Es de suponer que los lectores de este libro han estudiado concienzudamente el Manual y el libro de Introducción al ZX Spectrum, o las publicaciones correspondientes del Spectrum+, y ahora están deseando encontrar una guía paso a paso para aprender a escribir sus propios programas. Este libro es esa guía. Si se estudian los programas uno por uno en el orden en que aparecen en el libro, en muy poco tiempo se obtendrá un conocimiento mejor de la programación en BASIC.

El principio básico que recomendamos es estudiar este libro estando sentado enfrente del ordenador, con una unidad de cinta lista para grabar todo lo que se introduzca. De esta forma se asimilarán las instrucciones muy fácilmente.

Todos los programas fueron escritos primero en el ZX Spectrum. Entonces surgió la versión perfeccionada de ese ordenador con el nombre Spectrum+. Hubo que procesar los programas en este nuevo ordenador y todos pasaron perfec-

tamente. El Spectrum+ dispone de teclas especiales para la mayoría de las funciones del teclado, por lo que algunas de las explicaciones incluidas son ya innecesarias (la de la página 2 por ejemplo). Todas estas diferencias están perfectamente explicadas en la Guía del Usuario del ZX Spectrum+.

W. Simister

1. CREACION Y EMPLAZAMIENTO DE CARACTERES GRAFICOS

Una de las cosas que proporciona más satisfacciones cuando se comienza a dominar la escritura de programas en el ZX Spectrum consiste en generar un carácter gráfico de creación propia que se pueda colocar en cualquier lugar de la pantalla. Afortunadamente, estamos ante una de las cosas más fáciles de hacer, que sólo exige tres requisitos: (1) saber reducir una figura a una serie de números; (2) saber manejar el bucle FOR/NEXT, y (3) saber utilizar las instrucciones LOCATE/PRINT.

En primer lugar analizaremos el requisito (1), pero al hacerlo saldrán a colación los otros dos (aunque se explicarán más adelante). Introducir el programa siguiente; saldrá en la pantalla una ilustración a título de referencia. REM significa nota u observación (REMark en inglés); es ignorada por el ordenador pero resulta útil para el programador como refe-

rencia. Añadir los asteriscos (pulsando simultáneamente las teclas SYMBOL SHIFT y B).

```
 5 REM ** Generacion de graficos **
10 REM ****************************
15 LET X=40
20 FOR Y=24 TO 152 STEP 16
25 PLOT X,Y: DRAW 152,0
30 NEXT Y
35 FOR X=40 TO 180 STEP 16
40 LET Y=24
45 PLOT X,Y: DRAW 0,144
50 NEXT X
```

Una vez escritas estas líneas, pulsar RUN y luego ENTER. Producirán una configuración de cuadrados, ocho por ocho. Añadir ahora estas líneas.

```
55 PRINT OVER 1;AT 1,5;"1";AT 2,5;"286432
   16 8 4 2 1"
60 PRINT OVER 1;AT 4,21;"16";AT 6,21;"32"
   ;AT 8,21;"64";AT 10,21;"224";AT 12,21;
   "127";AT 14,21;"63";AT 16,21;"50";AT 1
   8,21;"82"
```

No omitir los espacios en blanco dejados entre los números 6 a 1 en la línea 55 porque son importantes. Al introducir las líneas de los programas, es verdaderamente esencial prestar especial atención a los signos de puntuación que separan letras o números en todas las líneas. La coma (,), el punto y coma (;), los dos puntos (:) y las comillas (") son signos que deben ser meticulosamente reproducidos tal como están en los listados. La línea 60, por ejemplo, tiene casi tantos signos de puntuación como letras y números, y todos cumplen una función. Si se deja de poner uno de ellos o se inserta uno donde no lo había, es probable que cuando se pulse ENTER la línea se niegue a subir a su sitio y aparezca en la citada línea un

signo de interrogación. Este signo de interrogación no siempre sale justo donde está el error (porque denuncia un error, no es una peculiaridad más del ordenador). Cuando esto ocurra, repasar todos los signos. Creemos que esta función de edición es una de las mejores prestaciones del Spectrum.

Al ejecutar el programa con RUN, las líneas 55 y 60 pondrán números en los bordes superior y lateral derecho de los cuadrados. Los números del borde superior corresponden a los valores de las columnas que encabezan. (Los números de la derecha tienen dedicada una explicación más adelante.) Ahora hay que introducir las líneas 65 a 110, pero vaya antes una pequeña explicación al respecto pues se puede cometer un error.

Los ochos representan cuadrados negros que salen impresos uno junto a otro; se introducen pulsando simultáneamente las teclas CAPS SHIFT y 9 (la tecla del modo gráfico) y luego las teclas SYMBOL SHIFT y 8 (que lleva grabado un cuadrado negro). Hay 2 cuadrados repetidos seis veces en la línea 65; 6 cuadrados repetidos dos veces en la línea 70; 14 cuadrados repetidos dos veces en la línea 75; 12 cuadrados repetidos dos veces en la línea 80; 4 cuadrados repetidos dos veces en la línea 85; 2 cuadrados repetidos tres veces en la línea 90, y 2 cuadrados repetidos tres veces en la línea 95.

Una vez introducido el número correcto de cuadrados, pulsar la tecla 9 sólo una vez; el ordenador saldrá del modo gráfico y por lo tanto se podrán seguir introduciendo cosas de la forma normal. Cada vez que se introduzca un carácter gráfico habrá que hacer esto. La REM A de la línea 105 tiene únicamente efectos informativos y es ignorada por el ordenador. La letra "A" de las líneas 100 y 110 debe ser introducida en el modo gráfico.

```
65 PRINT AT 3,11;"88";AT 4,11;"88";AT 5,9
   ;"88";AT 6,9;"88";AT 7,7;"88";AT 8,7;"
   88"
```

```
70 PRINT AT 9,5;"888888";AT 10,5;"88888"
75 PRINT AT 11,7;"88888888888888";AT 12,7
   ;"88888888888888"
80 PRINT AT 13,9;"888888888888";AT 14,9;"
   888888888888"
85 PRINT AT 15,9;"8888";AT 15,17;"88";AT
   16,9;"8888";AT 16,7;"88"
90 PRINT AT 17,7;"88";AT 17,11;"88";AT 17
   ,17 "88"
95 PRINT AT 18,7;"88";AT 18,11;"88";AT 18
   ,17 "88"
100 FOR X=0 TO 7: READ Y: POKE USR "A"+X,Y
   : NEXT X
105 DATA 16,32,64,224,127,63,50,82: REM A
110 PRINT AT 9,3;"A"
```

Grabar el programa antes de hacer ninguna otra cosa. Es ésta una muy buena costumbre que puede prevenir la pérdida de un tiempo de teclado precioso. Una vez que el programa está en cinta no importa que haya un corte de electricidad: se puede volver a cargar.

Ejecutar el programa ahora y saldrá en la pantalla una imagen similar a la de la figura 1.1.

Si se compara la línea 105 DATA con los números que hay en el lado derecho de la pantalla se verá de dónde proceden. Se introducen siempre así: con una coma entre cada número completo y ninguna al principio ni al final. Los dos puntos (:) sirven para separar la instrucción REM a fin de que el ordenador no piense que forma parte de la instrucción DATA.

Pero queda por saber de dónde han salido esos números. Ya sabemos que los números que hay encima de los 8 × 8 cuadrados corresponden a los valores de las columnas que encabezan. Estos números superiores son siempre (de derecha a izquierda): el 1; el 2 (el doble); el 4 (el doble); el 8 (el doble); el 16 (el doble); y así sucesivamente hasta el 128, que es el doble de 64. Es fácil recordar estos números.

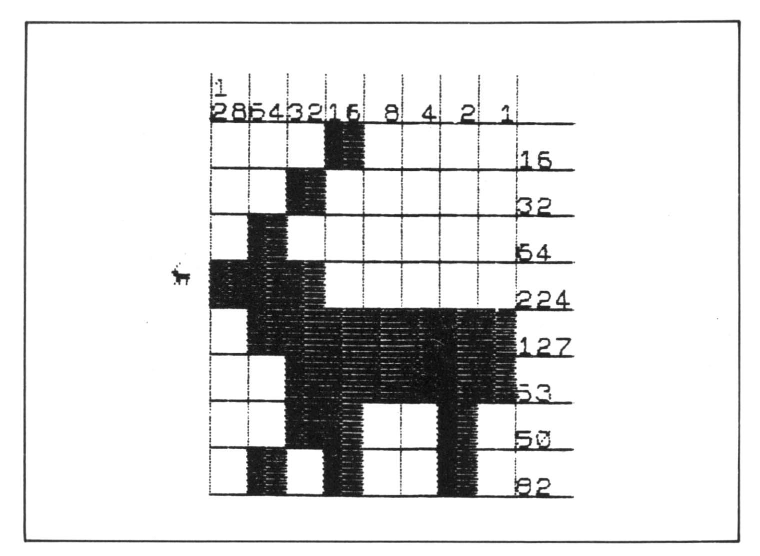


Figura 1.1

Dibujar en una hoja de papel un conjunto de 8 × 8 cuadrados vacíos, sin poner números en el lado derecho, pero incluyendo en el borde superior la serie de números doblados, del 1 al 128 comenzando por la derecha. Para dibujar un carácter gráfico basta rellenar a voluntad cuadrados hasta componer la figura que se quiera. Hagamos la figura de una oca.

Poner tres cuadrados negros en la primera fila, debajo de los números 128, 64 y 32. La suma de estos tres números da 224, y ése es precisamente el número a introducir para que vaya a parar a la derecha de la primera fila. La segunda fila tendrá un cuadrado negro en la columna 16, luego hay que introducir un 16 a la derecha de esa fila. La tercera fila tendrá un cuadrado en la columna 32, luego introducir ese número. La cuarta fila tiene cinco cuadrados negros, en las columnas 64, 32, 16, 8 y 4, luego hay que introducir en esa fila un 124. Después vienen los números 126, 63, 15 y 4, a introducir en las siguientes cuatro filas.

Tres nuevas líneas bastarán para incluir estos nuevos números:

```
115 FOR X=0 TO 7: READ Y: POKE USR "B"+X,Y
    : NEXT Y
120 DATA 224,16,32,124,126,63,15,4: REM
125 PRINT AT 11,3;"B"
```

Al ejecutar este programa se verá una oca justo debajo del pequeño ciervo de antes. Estos dos caracteres gráficos permanecerán en el ordenador hasta que se desconecte. Ni siquiera la orden NEW los afectará.

El análisis de los resultados obtenidos puede hacer pensar que con una hoja de papel cuadriculado se puede diseñar y reproducir casi cualquier tipo de carácter, pero hay una limitación: sólo se pueden hacer 22 caracteres. A estos caracteres se pueden asignar las teclas de la A a la U, pero ninguna más por ahora.

Ha llegado el momento de añadir otra línea: 295 STOP. Hay que ponerla porque vamos a pasar a otra parte del programa (a la línea 300) y no queremos que sea ejecutada junto con esta primera parte. De todas formas, queda suficiente espacio entre las líneas 125 y 295 para que cada uno escriba las líneas que quiera, a título de ejercicio, y las añada al programa. Grabar (SAVE) todo lo hecho hasta ahora y luego escribir:

```
300 REM ** PRINT AT **
305 REM **************
310 BORDER 5: CLS
315 LET X=0
320 FOR Y=0 TO 175 STEP 8
325 PLOT X,Y: DRAW 255,0
330 NEXT Y
335 FOR X=0 TO 255 STEP 8
340 LET Y=0
```

```
345 PLOT X,Y: DRAW 0,175
350 NEXT X
355 PRINT OVER 1;AT 0,0;"0123456789&hasta-
    el-31"
360 PRINT OVER 1;AT 1,0;"1";AT 2,0;"2";AT
    3,0;"3";AT 4,0;"4";AT 5,0;"5";AT 6,0;"
    6";AT 7,0;"7";AT 8,0;"8";AT 9,0;"9";AT
    10,0;"&";AT 11,0;"h";AT 12,0;"a"
365 PRINT OVER 1;AT 13,0;"s";AT 14,0;"t";A
    T 15,0;"a";AT 16,0;"-";AT 17,0;"e";AT
    18,0;"l";AT 19,0;"-";AT 20,0;"2";AT 21
    ,0;"1"
```

Repasar todas las entradas prestando especial atención a los signos de puntuación (especialmente en las líneas 360 y 365), y no confundir el 0 (cero) con la O (letra O). Grabarlo en cinta junto con el primer programa y luego introducir RUN 300 y pulsar ENTER. Esta acción pondrá en la pantalla un borde azul que define el área que el usuario puede utilizar en condiciones normales. La porción central blanca será una cuadrícula de cuadrados.

Hay números en el borde superior (0 al 31) y en el lateral derecho (0 al 21). Estos últimos no se distinguen muy bien pero sí suficiente para nuestros propósitos. Hay en total 704 posiciones (32 × 22 porque el 0 cuenta como un número más). En cada posición se puede imprimir un carácter de cualquier tipo, siempre y cuando sea del tamaño de un solo espacio; las letras y los números ocupan todos sólo un espacio. La orden OVER 1 puesta justo a continuación de PRINT permite imprimir ahí letras sin hacer desaparecer por ello las líneas dibujadas previamente. Veamos qué queremos decir con esto.

Escribir LIST 360 y después pulsar ENTER. Saldrá en la pantalla una parte del programa. Algunas veces sale también al pie de la pantalla la palabra SCROLL (corrimiento). Si sale esa palabra, pulsar la N para impedir que el programa corra por la pantalla hacia arriba.

Se verá una flecha (el *cursor*) ubicada en la línea 360 del programa. Pulsar a la vez las teclas CAPS SHIFT y 1 (tecla EDIT) y la línea 360 aparecerá en el pie de la pantalla; estando aquí se puede EDITar. Manteniendo CAPS SHIFT apretada mientras se pulsa repetidamente la tecla 8 (que lleva una flecha encima), se moverá el cursor a la derecha (si va demasiado lejos, una pulsación de la tecla 5 le devolverá hacia atrás). Partiendo de la posición original, cuatro pulsaciones de la tecla 8 colocarán el cursor entre OVER 1; y AT.

Manteniendo un dedo apretado sobre la tecla CAPS SHIFT, pulsar tres veces la tecla 0 (que lleva la palabra DELete encima). Soltar CAPS SHIFT y pulsar ENTER. La línea 360 volverá a ocupar su sitio en el programa pero ahora no estará OVER 1. Escribir RUN 300 y pulsar luego ENTER.

En la pantalla se verá que en el lado izquierdo están borradas algunas líneas que pasan entre números. Una vez visto esto, para volver a poner OVER 1 hay que bajar otra vez la línea 360, mover el cursor a la derecha pulsando una vez la tecla correspondiente (quedará justo a continuación de PRINT) y luego escribir OVER;, sin olvidar el punto y coma (;).

A estas alturas, el lector ya se habrá dado cuenta que OVER es una palabra reservada, que se obtiene pulsando simultáneamente las teclas CAPS SHIFT y SYMBOL SHIFT y manteniendo luego pulsada SYMBOL SHIFT mientras se pulsa la tecla N para componer OVER. Devolver la línea 360 al programa (pulsando ENTER) e introducir otra vez RUN 300 a efectos de referencia.

Hay 704 cuadrados disponibles para poner caracteres mediante instrucciones PRINT AT seguidas por las coordenadas X, Y. La X corresponde a los números del lateral izquierdo, los números de fila, y la Y corresponde a los números del borde superior, los números de columna. Se trata simplemente de una convención que facilita las cosas. Estas órdenes

tienen siempre la forma siguiente: PRINT AT (número lateral izquierdo) (,) (número borde superior) (;) y luego (”) antes y después de lo que se quiere imprimir. Hay algunas excepciones con respecto a las comillas, pero más tarde hablaremos de ellas. A título de demostración introducir:

```
370 PRINT AT 2,5;"A"
```

“A” se introduce en el modo gráfico. RUN 300 y ahí estará, en la intersección de la fila 2 con la columna 5, el pequeño ciervo. No olvidar que aunque ahora estemos introduciendo RUN 300, los caracteres gráficos asociados a la A y a la B siguen en la memoria del ordenador. Para confirmar lo dicho, y para mostrar lo fácil que es trabajar con la instrucción PRINT AT, introduzcamos ahora esta otra línea (pulsar A y B estando en el modo gráfico):

```
375 PRINT AT 4,5;"B";AT 4,7;"AB"
```

AT hace las mismas funciones que PRINT AT en la misma línea. Más adelante volveremos a tocar este punto. De ahora en adelante, cuando digamos RUN o ejecutar sin más queremos referirnos sólo a la sección en la que se está trabajando (en este caso es RUN 300). Habiendo introducido y visto el efecto de la línea 375, añadir ahora estas dos líneas más:

```
380 PRINT AT 6,2;"Esta es una prueba de es
    critura"
385 PRINT OVER 1;AT 8,2;"Esta es una prueb
    a de escritura"
```

Cuando se ejecutan estas líneas se ve la diferencia entre poner OVER 1 y no ponerlo. Las dos líneas son iguales, con excepción del diferente emplazamiento y de la presencia o ausencia de OVER 1. La figura 1.2 muestra los resultados de ambas.

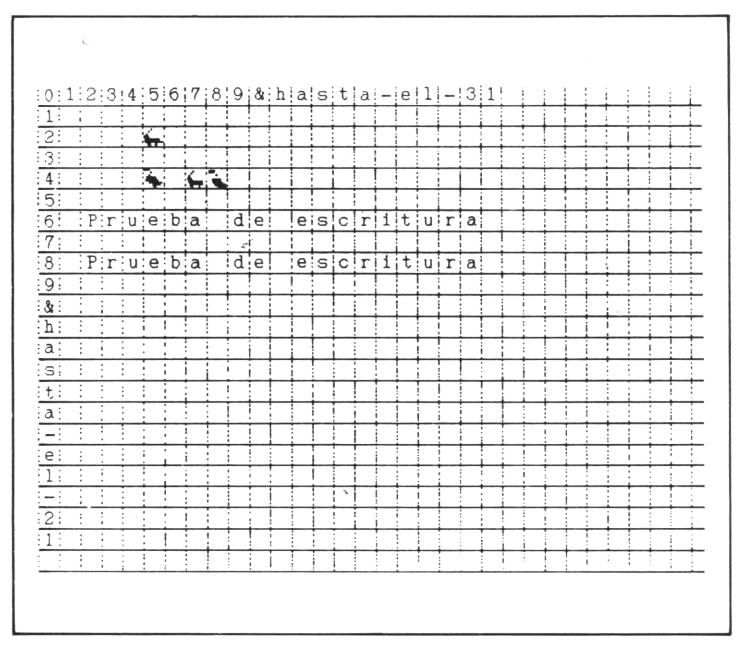


Figura 1.2

Esta demostración debe ser suficiente para que el lector se atreva a añadir otras líneas de su invención a continuación de la línea 385. Hacerlo como práctica y, cuando se haya acabado, añadir la línea final 496 STOP porque ahora vamos a ver los bucles FOR/NEXT.

Estudiar antes las líneas 20 a 50 de la primera sección. Están constituidas por dos bucles FOR/NEXT que gobiernan la impresión de líneas en la pantalla. Además, la línea 100 es un bucle FOR/NEXT completo y la línea 115 también. Aunque utilizados con distintos fines, los bucles siguen siempre una fórmula predeterminada: FOR va seguida siempre por una letra (normalmente X), luego va el signo = (igual) y luego (algo) TO (algo). Las órdenes siguientes dicen lo que tiene que hacer el bucle. Por último viene NEXT X (o la letra

puesta a continuación de FOR). Introducir estas cinco líneas (A en el modo gráfico):

```
500 REM ** Bucles FOR/NEXT **
505 REM *********************
510 FOR X=2 TO 18: LET Y=2
515 PRINT AT X,Y;"A"
520 NEXT X
```

Introducir RUN 300 y saldrá una columna de ciervos a lo largo del lado izquierdo de la pantalla. Introducir ahora estas otras tres líneas, que son iguales que las anteriores excepto por un par de diferencias. Que el lector trate de descubrirlas.

```
525 FOR X=2 TO 18 STEP 2: LET Y=4
530 PRINT AT X,Y;"A"
535 NEXT X
```

El lector habrá visto que Y es ahora 4, y por eso sale impreso el resultado de estas líneas en la columna 4. También habrá descubierto que se ha añadido STEP 2 a la parte FOR del bucle; es ese STEP 2 el que separa las figuras. Hacer una prueba con estas otras cuatro líneas (introduciendo A en el modo gráfico):

```
540 FOR X=2 TO 18 STEP 2
545 FOR Y=6 TO 28 STEP 2
550 PRINT AT X,Y;"A"
555 NEXT Y: NEXT X
```

Antes de seguir adelante, grabar en cinta todos los programas introducidos hasta ahora. Una vez grabados y verificados, introducir una vez más RUN 500. El resultado será una pantalla llena de ciervos, siempre y cuando no se haya olvidado introducir todas las "A" en el modo gráfico. Para explicar por qué se ha llenado la pantalla de ciervos hay que hacer un análisis detenido de la situación.

Hay cuatro bucles FOR/NEXT entre las líneas 510 a 555; el último está entre las líneas 540 y 555, donde el bucle Y está encerrado por el bucle X (se dice que está “anidado” en el bucle X; más adelante seguiremos hablando sobre el anidamiento de bucles). Comenzando por la instrucción FOR X=2 TO 18 de la línea 510 y acabando por la instrucción NEXT X de la línea 520, el ordenador lee X=2 TO 18 y sabe entonces que lo que se le ordene que haga debe hacerlo en todas las filas comprendidas entre la 2 y la 18. Ejecuta las líneas de instrucciones siguientes hasta que se encuentra con NEXT X, y entonces vuelve a FOR para ejecutar esas líneas otra vez. Repite este bucle hasta haber utilizado todas las filas comprendidas entre la 2 y la 18, y luego pasa a la siguiente línea.

Ahí, en la línea 555, se encuentra con otra orden FOR y ve que esta vez debe utilizar filas alternas: 2, 4, 6, 8, etc. porque así lo ordena STEP 2. Ve también que debe utilizar siempre la columna 4 pues ahora Y=4. Así lo hace. NEXT X le envía de vuelta a la línea 525 para que haga todo otra vez, y así sucesivamente hasta que llega a la fila 18, en cuyo momento, habiendo completado la instrucción 2 TO 18 STEP 2, pasa a la línea 540.

Aquí se encuentra con la misma línea X, pero ahora Y forma parte también de una orden FOR: FOR Y=6 TO 28 STEP 2. En consecuencia, procede a imprimir su primera A gráfica en la fila 2 (FOR X) y en las columnas 6, 8, 10, etc. (FOR Y). Luego se encuentra otra vez con NEXT X y vuelve por lo tanto a la línea 540 para hacer otra vez lo mismo, pero ahora en la fila 4. Y así sigue hasta haber completado todas las órdenes 2 TO 18 STEP 2. Entonces ve que no hay nada más que hacer e imprime “0 OK, 555:2” para manifestar que ha acabado.

El último bucle recibe el nombre de bucle anidado. Es imperativo decir que todos los bucles que actúen al mismo tiempo deben estar organizados de forma que cada uno esté totalmente encerrado o anidado en otro: la letra que sigue a

la primera FOR debe ser la misma que sigue a la última NEXT. Permitir que los bucles se monten parcialmente es no dejar que el ordenador ejecute las órdenes impartidas.

El empleo del bucle FOR/NEXT para generar un carácter gráfico sigue una pauta similar, pero lo que hay que hacer ahora es poner 8 números en la memoria de forma que el ordenador pueda recuperar todo el paquete de 8 números cuando así se le ordene. Aquí es necesario recurrir a una explicación un poco más intrincada. Utilizaremos las líneas 100 a 110 como referencia.

100 FOR X=0 TO 7. Hay ocho filas (de ocho pixels cada una) en un carácter gráfico, y para el ordenador la 0 es una fila más. READ Y. Esa es la posición Y de la memoria del ordenador, pues la memoria tiene una organización interna que elige automáticamente un espacio de la memoria para los caracteres gráficos, con independencia de la letra que decidamos poner tras READ. READ es la palabra operativa.

POKE USR "A"+X, Y dice al ordenador que coja el primer número de DATA y que lo ponga en la primera línea de Y (etiquetando esta posición a sus efectos particulares como "A" en modo gráfico). Luego llega a NEXT X, que le envía de vuelta a FOR, tras lo cual coge el segundo número de DATA. Y así sigue hasta la octava línea, en cuyo momento ha completado esa tarea y puede pasar a hacer otra.

Los bucles FOR/NEXT se pueden utilizar también para generar una serie de pitidos (BEEP en inglés). Introducir:

```
560 FOR X=0 TO 20: BEEP .1,X: NEXT X
```

Si se introduce ahora RUN 500 se oirá al final una serie ascendente de notas, 21 en total, pues NEXT X dice al ordenador que vuelva y toque otra nota hasta recorrer todos los números 0 TO 20. Quien quiera hacer prácticas con esta prestación, que no olvide que la orden BEEP lleva dos parámetros

separados por una coma: el primero es la duración de la nota y el segundo es el tono o frecuencia de la nota. Obedeciendo a la línea 560, el ordenador toca primero una nota de tono 0 durante 0,1 segundo, luego vuelve atrás y toca otra nota de tono 1 y la misma duración, luego otra de tono 2, y así hasta 21. Como ésta es la última línea, cuando el ordenador la complete presentará en pantalla el aviso OK usual.

Los bucles FOR/NEXT no sirven para tocar una melodía pues las variaciones de tono y duración son demasiado diversas, pero sí sirve para repetir una parte. Se hace mediante una modificación de la línea 560 consistente en añadir otro bucle FOR/NEXT alrededor del ya programado. Esa línea queda así:

```
560 FOR A=0 TO 3: FOR X=0 TO 20: BEEP.1,X+
    A: NEXT X: NEXT A
```

La forma más fácil de modificar la línea 560 consiste en traerla al pie de la pantalla para EDITarla tal como se ha explicado anteriormente (con LIST 560, etc.). Tras el número 560, añadir FOR A=0 TO 3:, mover el cursor hasta la posición adecuada para escribir +A a continuación de la X de la orden BEEP, y añadir luego: NEXT A al final.

Hasta aquí hemos dado una explicación sencilla del anidamiento y forma de empleo de los bucles FOR/NEXT. Obsérvese que la X dice al ordenador qué tiene que tocar mientras que la A le dice cuántas veces debe hacerlo. El resultado de RUN 500 es la figura 1.3.

Quien haya seguido paso a paso todas las instrucciones de este capítulo tendrá escrito un programa compuesto por tres demostraciones independientes. Sería bueno poder pasar de una sección a otra a voluntad, y se puede hacer añadiendo unas pocas líneas. Constituyen una secuencia INPUT que añadiremos a la primera parte, sección 1, borrando la línea 295 STOP.

Figura 1.3

```
285 INPUT "Pulsar 2 para seccion siguiente
    ";A$
290 IF A$<>"2" THEN GO TO 285
295 IF A$="2" THEN CLS: GO TO 300
```

El signo no igual de la línea 290 está en la tecla W. Una vez escritas estas líneas, al introducir RUN (sin número de línea) saldrá en pantalla el primer programa, y debajo de él estará el mensaje INPUT "Pulsar 2 para sección siguiente". Si se introduce el 2, la pantalla se borrará y saldrá la siguiente sección.

Las órdenes INPUT introducidas de esta forma salen siempre en el pie de la pantalla, fuera del área principal de la pantalla, acompañadas por un cursor intermitente constituido por el signo de las comillas; el 2 introducido va a parar ahí. Al pulsar ENTER cambia la pantalla, bajo el control de la línea 295. La línea 290 está ahí haciendo las funciones de vigilante

que devuelve el ordenador a la línea 295 cuando el usuario introduce inadvertidamente otra letra o número. Hacer la prueba y se verá que la INPUT sigue donde estaba. CLS dice al ordenador que borre todo lo que haya en la pantalla en ese momento.

Damos ahora las líneas a introducir para las otras dos secciones. Aunque algunas empiezan por 4 y otras por 5, no hay ninguna posibilidad de error achacable al ordenador; a medida que se pulsa ENTER, la línea va al sitio del programa que le corresponde.

```
485 INPUT "Pulsar 3 para seccion siguiente
    ";B$
490 IF B$<>"3" THEN GO TO 485
495 IF B$="3" THEN CLS: GO TO 400
585 INPUT "Pulsar 1 para seccion primera";
    C$
590 IF C$<>"1" THEN GO TO 585
595 IF C$="2" THEN CLS: GO TO 300
```

Una vez introducidas estas líneas y ejecutado el programa para comprobar que todo está bien y funciona correctamente, grabar todo en cinta.

Una cosa más: Cuando se ejecuta el programa y sale al pie de la pantalla el mensaje INPUT, puede parecer que no hay forma alguna de volver al listado. Se puede hacer de esta forma: pulsar CAPS SHIFT y EDIT simultáneamente. Esta acción quitará el cursor de la instrucción INPUT. Pulsando STOP y luego ENTER saldrá al pie de la pantalla un mensaje "STOP IN INPUT". Ahora se puede hacer lo que se quiera, como LIST o RUN.

2. UN JUEGO DE MESA

Si se ha estudiado y completado concienzudamente el primer capítulo, no hay nada que se interponga entre el deseo de escribir un programa de juegos y su realización, aparte de reflexionar bastante y seguir pacientemente las instrucciones de este capítulo.

Se ha elegido como ejemplo un solitario con fichas de dominó, básicamente porque permite poner en práctica muchas de las cosas aprendidas hasta ahora, y además porque permite utilizar muchas más fichas que un juego de dominó normal.

Un programa de estas características tendrá que tener muchas secciones. Se necesitará una para generar las fichas de dominó y otra para desarrollar el juego. Además, sería bueno tener una pantalla con las reglas del juego antes de comenzar a jugar, por lo que habrá que incluir una sección a estos efectos, y otra para componer la pantalla del juego real. Comenzaremos generando las fichas de dominó.

Una vez introducidas en el ordenador las instrucciones necesarias para que genere caracteres gráficos correspondientes a las fichas de dominó, éstas quedan almacenadas en la memoria, a entera disposición del ordenador para su utilización durante el transcurso de sucesivas partidas. No es necesario

por lo tanto que esta sección esté incluida en la sección del juego: el ordenador tendría que ejecutarla cada vez que se juega una partida e invertiría en hacerlo un tiempo que por corto que fuera introduciría un retardo innecesario.

Por consiguiente, utilizaremos una orden llamada GOSUB/RETURN que nos permitirá situar la sección de generación de gráficos al final del programa, a partir de la línea 5000. Para acceder a ella incluiremos al principio del programa una línea GO SUB 5000. Escribir ahora las siguientes líneas:

```
  10 REM SOLITARIO CON FICHAS DE
     DOMINO
  20 BORDER 5: PAPAER 7: CLS
  30 POKE 23609,255: POKE 23658,8
  40 GO SUB 5000
4999 STOP
5000 REM DISEÑO DE LAS FICHAS
5999 RETURN
```

Las letras en vídeo inverso que siguen a REM se obtienen pulsando simultáneamente las teclas CAPS SHIFT y 4. Para volver a las letras normales pulsar simultáneamente CAPS SHIFT y 3. En primer lugar vamos a levantar un esqueleto y a él iremos incorporando "la carne" del programa. Se necesita una sección para las reglas, otra para componer la pantalla del juego y otra para las acciones reales. A estos efectos, introducir ahora estas líneas:

```
  50 GO SUB 4000
  60 GO SUB 3000
 100 REM JUEGO
2999 STOP
3000 REM PANTALLA DEL JUEGO
3998 RETURN
3999 STOP
4000 REM REGLAS
4998 RETURN
```

Estas líneas ocuparán en el listado los lugares que, por sus números, les corresponden. Al estudiar este listado se ve claramente que constituye un buen marco para el programa, aunque también es cierto que es preciso dar algunas explicaciones. En la línea 30, la instrucción POKE 23609,255 hace que el ordenador emita un pitido (BEEP) cada vez que se pulsa una tecla, una prestación de gran utilidad a la hora de introducir algo por el teclado. La otra POKE pasa el ordenador al modo MAYUSCULAS. La línea 40 GOSUB 5000 envía el ordenador a la línea 5000, y la línea 5999 RETURN lo envía de vuelta a la línea inmediatamente posterior a la GO SUB que lo envió a la línea 5000: el ordenador ni se olvida ni se equivoca. Por consiguiente, la línea 50 GO SUB 4000 envía el ordenador a la sección REGLAS, donde la 4998 RETURN lo devuelve a la línea siguiente a la 50, y así sucesivamente. Las instrucciones STOP de las líneas 2999, 3999 y 4999 han sido puestas ahí por conveniencias del momento; esas líneas evitan que el ordenador entre en una de las secciones siguientes en caso de que se hubiera dejado de poner una RETURN en alguna sección. Ahora que tenemos un buen esqueleto, comencemos a trabajar en cada sección por turnos, comenzando por la 5000.

Utilizaremos caracteres gráficos para hacer las fichas de dominó; las fichas serán de color negro y los puntos de color blanco. No obstante, como las fichas de dominó son de forma rectangular mientras que los caracteres gráficos son cuadrados, se necesitarán dos gráficos para componer una ficha de dominó, lo que eleva a catorce el número total de caracteres necesarios. Nuestra primera línea es por lo tanto: 5010 FOR X=0 TO 13 (el cero cuenta como uno), seguida por FOR=0 TO 7: READ Z: POKE USR CHR$(144+X)+Y,Z: NEXT Y: NEXT X. No introducir esto todavía.

¿Alguien no sabe de dónde sale ese CHR$? En el capítulo 1 utilizábamos las letras A & B en el modo gráfico en esa frase POKE USR. Pues bien, CHR$ 144 es el código (CODE) de esa A, por lo que la X sumada a 144 (144+X) está diciendo

al ordenador que utilice las teclas ABCDEFGHIJKLMN para los caracteres gráficos que vamos a componer. La instrucción FOR Y=0 TO 7 dice que cada carácter tendrá ocho líneas, luego esa línea única (5010) es una instrucción completa.

Ahora hay que diseñar las fichas de dominó. Como un bloque lleno de unos (el cuadrado 8 × 8 lleno) puesto junto a otro bloque similar, sin separación entre ambos, compone un bonito rectángulo, entre los dos reproducen perfectamente la ficha doble blanca. Sin embargo, si se imprimen otros dos bloques justo a continuación de los dos primeros, no habrá separación entre las dos parejas y no compondrán por lo tanto dos dobles blancas. Esta es la razón de que la media ficha izquierda de cada pareja deba tener una línea de ceros en su primera columna (la columna 128) con objeto de que defina una pequeña separación o hueco. Por eso es preciso definir catorce caracteres para representar las siete fichas de dominó (0 a 6): dos unos, dos doses, etc.

Siguiendo el procedimiento bosquejado en el capítulo 1, dibujar un cuadrado 8 × 8 e introducir ahí la primera media ficha de dominó, un seis para empezar por la de la izquierda. Poner un cero en todos los primeros espacios (la columna 128), dejando por definir siete espacios en cada fila. Afortunadamente, en estos siete espacios caben cuatro unos y tres ceros, luego es posible meter tres puntos blancos sobre un fondo negro. Sin embargo, un punto único no se apreciaría con suficiente claridad, por lo cual hay que utilizar una fila más a estos efectos, la inferior a aquélla. Llenar la primera fila (de las siete que quedan) con unos; la segunda y tercera filas con unos y ceros alternados (1010101); la cuarta y quinta filas con unos; la sexta y séptima filas igual que la segunda y la tercera; y la última fila con unos.

Quien quiera puede diseñar las demás fichas de dominó una por una, pero si se han entendido bien los principios fundamentales de la composición de los caracteres gráficos, basta copiar directamente las líneas DATA 5020 a 5050:

```
5010 FOR X=0 TO 13: FOR Y=0 TO 7: READ Z: P
     OKE USR CHR$(144+X)+Y,Z: NEXT Y: NEXT
     X
5020 DATA 127,127,127,119,119,127,127,127,1
     27,125,125,127,127,95,95,127,127,125,1
     25,119,119,95,95,127,127,93,93127,127,
     93,93,127: REM AABBCCDD
5030 DATA 127,93,93,119,119,93,93,127,127,8
     5,85,127,127,85,85,127,127,127,127,127
     ,127,127,127,127: REM EEFFGG
5040 DATA 255,255,255,247,247,255,255,255,2
     55,253,253,255,255,223,223,255,255,253
     ,253,247,247,223,223,255,255,221,221,2
     55,255,221,221,255:REM HHIIJJKK
5050 DATA 255,221,221,247,247,221,221,255,2
     55,213,213,255,255,213,213,255,255,255
     ,255,255,255,255,255,255:REM LLMMNN
```

Se observará que al final de cada fila DATA hay una instrucción REM: AABBCCDD etc. La primera letra de cada pareja (AA) se introduce en el modo MAYUSCULAS, pero la segunda se introduce en el modo gráfico (CAPS SHIFT y 9). Cuando se introduce así la instrucción REM, en el listado que sale en la pantalla una vez dada la orden RUN para que queden almacenados los caracteres en la memoria, sale tras la REM una letra y su carácter asociado, lo cual sirve para recordar la letra a la que se refiere esa DATA.

Una vez introducida la lista, ejecutar el programa (RUN). En la pantalla no se verá nada, con excepción del aviso OK del pie. Introducir ahora LIST y hacer uso de la prestación SCROLL (corrimiento) para ver las últimas líneas. Si todo ha sido hecho correctamente, cada instrucción REM mostrará una media ficha de dominó junto a su letra. En caso contrario, repasar todo lo introducido.

Habiendo completado la sección de los caracteres gráficos, sección 5000, pasamos a las REGLAS, sección 4000. Esta sec-

ción presenta en pantalla la posición que ocuparán las fichas en el transcurso del juego, luego es necesario estudiar una vez más la instrucción PRINT AT.

En el lado izquierdo de la pantalla están los números X (del 0 al 21 en orden descendente) y a lo largo de la cabecera de la pantalla están los números Y (del 0 al 31). Los números mayores que el 9 ocuparían dos espacios por número, razón por la que ha llegado a ser usual definir las posiciones de la pantalla con letras que representen a esos números. Definiremos siete letras extendidas horizontalmente a lo largo de la cabecera de la pantalla y siete extendidas verticalmente a lo largo del lateral izquierdo, con una separación de dos espacios entre cada una de forma que en cada división se pueda colocar una ficha de dominó y un espacio. Las dos primeras líneas serán la 4010 y la 4020. Introducirlas:

```
4010 PRINT AT 0,1;"A  B  C  D  E  F  G";AT
     1,0;"A";AT 4,0;"B";AT 7,0;"C";AT 10,0;
     "D";AT 13,0;"E";AT 16,0;"F";AT 19,0;"G
     "
4020 LET A=1: LET B=4: LET C=7:LET D=10: LE
     T E=13: LET F=16: LET G=19
```

La línea 4020 LET A=1: LET B=4: etc. hace que, después de leer esa línea, el ordenador acepte las letras como equivalentes de los parámetros X,Y en las instrucciones PRINT AT. Por ejemplo, A,E significará 1,13 y definirá esa posición X,Y. Hacer este ejercicio: introducir (sin número de línea) PRINT AT D,C;“AA”; saldrán dos A en esa posición D,C. Este ejercicio no afectará en absoluto al programa y las A desaparecerán al hacer la siguiente entrada.

Ahora es preciso informar al ordenador qué ficha debe colocar en su posición cuando se le ordene, y para ello se utiliza una “CADENA”. Lo hacemos con LET A$ (pulsación simultánea de las teclas SYMBOL SHIFT y 4)=“ ”. Lo escrito entre las comillas es la cadena, donde cada letra o número

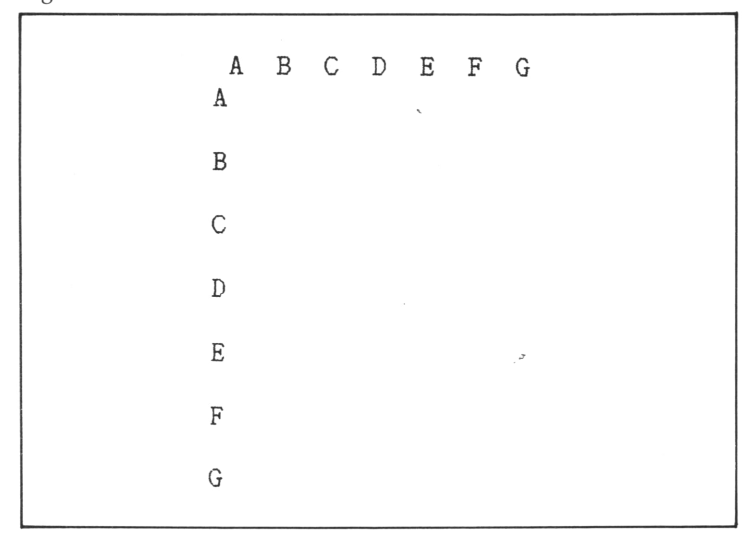
tiene una posición determinada. Si ha de haber siete caracteres, se pueden direccionar así: A$(1), A$(5), etc. En consecuencia, las dos líneas siguientes serán:

```
4030 LET A$="ABCDEFG"
4040 LET B$="HIJKLMN"
```

En estas dos líneas, las letras A a la N deben ser introducidas en el modo gráfico. Una vez escritas, ejecutar todo el programa. El resultado debe ser la aparición en pantalla de la figura 2.1.

A lo largo de la cabecera de la pantalla sale una fila de letras A B C, etc., y a lo alto del lateral izquierdo sale la misma columna de letras. Si se introduce LIST, el listado del programa mostrará claramente todos los caracteres gráficos de las fichas de dominó. Ahora hay que colocarlas en las posiciones elegidas con órdenes como PRINT AT A,A; A$(6); B$ (6). Este es uno de los casos en que no se ponen comillas

Figura 2.1



tras las coordenadas X, Y en una instrucción PRINT AT. Introducir ahora las siguientes líneas:

```
4050 PRINT AT A,A;A$(6);B$(6);AT A,B;A$(5);
     B$(5);AT A,C;A$(4);B$(4);AT A,D;A$(3);
     B$(3);AT A,E;A$(2);B$(2);AT A,F;A$(1);
     B$(1);AT A,G;A$(7);B$(7)
4060 PRINT AT B,A;A$(6);B$(5);AT B,B;A$(5);
     B$(4);AT B,C;A$(4);B$(3);AT B,D;A$(3);
     B$(2);AT B,E;A$(2);B$(1);AT B,F;A$(1);
     B$(7)
4070 PRINT AT C,A;A$(6);B$(4);AT C,B;A$(5);
     B$(3);AT C,C;A$(4);B$(2);AT C,D;A$(3);
     B$(1);AT C,E;A$(2);B$(7)
4080 PRINT AT D,A;A$(6);B$(3);AT D,B;A$(5);
     B$(2);AT D,C;A$(4);B$(1);AT D,D;A$(3);
     B$(7);AT E,A;A$(6);B$(2);AT E,B;A$(5);
     B$(1);AT E,C;A$(4);B$(7)
4090 PRINT AT F,A;AS(6);BS(1);AT F,B;AS(5);
     BS(7);AT G,A;AS(6);BS(7);AT 19,8;"X"
```

No dejar de estudiar estas líneas hasta entender perfectamente lo que hacen porque contienen un conjunto de instrucciones de primordial importancia. La línea 4020 daba un valor numérico a las letras A a la G y las líneas 4030 y 4040 ponían las dos mitades de las fichas de dominó en cadenas independientes. Pues bien, ahora las líneas 4050 a 4090 dicen al ordenador que las imprima en los sitios especificados. PRINT AT A,A; A$(6) significa: imprimir en A (posición X=1), A (posición Y=1) A$(6) (la sexta porción de la cadena A) junto a B$(6) (la sexta porción de la cadena B).

Cuando se haya entendido bien todo esto y esté todo introducido, ejecutar el programa. El resultado debe ser la aparición en pantalla de la figura 2.2. En caso contrario repasar todas las entradas hechas hasta ahora.

Se verá que en A,A está la ficha doble 6, en A,B la doble 5, en B,A la 6-5, y así sucesivamente. Esta es la forma final

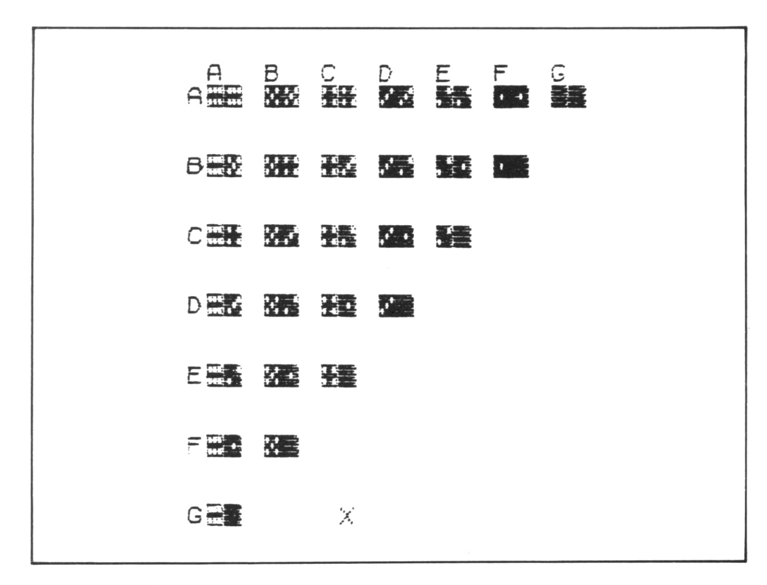


Figura 2.2

del juego “SOLITARIO DE DOMINO” que estamos componiendo. Esa pantalla está ahí para mostrar las REGLAS del juego (que se introducirán a continuación). Tanto las reglas como la pantalla desaparecerán al comenzar a jugar. La X ha sido puesta ahí para mostrar la posición donde estará situada la ficha elegida al azar por el juego. Las siguientes entradas son comparativamente directas.

Introducir ahora estas líneas:

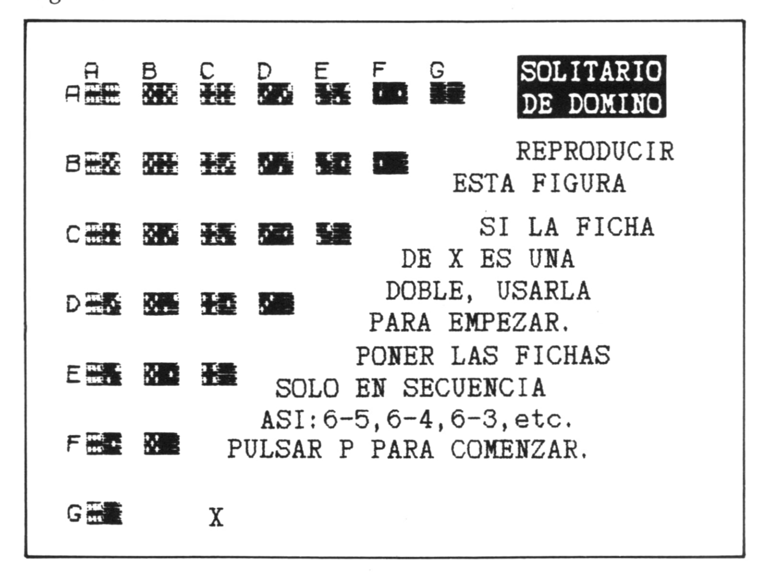
```
4100 PRINT INVERSE 1;AT 1,22;"SOLITARIO";AT
     2,22;"DE DOMINO"
4110 PRINT AT 4,20;"REPRODUCIR";AT 5,20;"ES
     TA FIGURA";AT 7,18;"SI LA FICHA";AT 8,
     17;"DE X ES UNA";AT 9,16;"DOBLE, USARL
     A";AT 10,15;"PARA EMPEZAR."
4120 PRINT AT 12,13;"PONER LAS FICHAS";AT 1
     3,12;"SOLO EN SECUENCIA";AT 14,11;"ASI
     :6-5,6-4,6-3,etc"
```

```
4130 PRINT AT 16,9;"PULSAR P PARA COMENZAR"
4140 INPUT "PULSAR P PARA COMENZAR:"Q$
4150 IF Q$<>"P" THEN GO TO 4140
4160 IF Q$="P" THEN CLS: RETURN
```

Obsérvese la inclusión de INVERSE 1 en la línea 4100. Este es el médoto que se utiliza normalmente para que en la pantalla salgan imprésos los caracteres en vídeo inverso (en el programa propiamente dicho lo hacíamos con CAPS SHIFT y 4). El mensaje INPUT del final saldrá en el pie de la pantalla y esperará hasta que se introduzca la "P" para borrar esa pantalla y poner la pantalla del juego, figura 2.3.

Habiendo completado las secciones 5000 y 4000, es el momento de pasar a la 3000, la composición de la pantalla del juego. Hay que indicar de alguna forma la posición donde deben ir a parar las fichas seleccionadas, a cuyo efecto hay que

Figura 2.3



hacer uso de las letras X,Y (ABCD etc.). Por lo tanto, las primeras cuatro líneas son:

```
3010 PRINT AT 0,1;"A  B  C  D  E  F  G";AT
     1,0;"A";AT 4,0;"B";AT 7,0;"C";AT 10,0;
     "D";AT 13,0;"E";AT 16,0;"F";AT 19,0;"G
     "
3020 LET A=1: LET B=4: LET C=7: LET D=10: L
     ET E=13: LET F=16: LET G=19
3030 PLOT 1,152: DRAW 171,0: PLOT 1,128: DR
     AW 147,0: PLOT 1,104: DRAW 123,0: PLOT
     1,80: DRAW 99,0: PLOT 1,56: DRAW 75,0:
     PLOT 1,32: DRAW 51,0: PLOT 1,8: DRAW 2
     7,0
3040 PLOT 28,172: DRAW 0,-164: PLOT 52,172:
     DRAW 0,-140: PLOT 76,172: DRAW 0,-116:
     PLOT 100,172: DRAW 0,-92: PLOT 124,172
     : DRAW 0,-68: PLOT 148,172: DRAW 0,-44
     : PLOT 172,172: DRAW 0,-20
```

Las líneas 3010 y 3020 son copias de las líneas 4010 y 4020, luego se pueden obtener editando aquéllas: LIST 4010 (y ENTER) hará aparecer esa línea en la pantalla junto con un cursor que señala el número 4010. Pulsando simultáneamente las teclas CAPS SHIFT y 1 (EDIT) se trae toda la línea al pie de la pantalla. Una vez ahí, moviendo el cursor tres espacios a la izquierda (CAPS SHIFT más 5) se puede borrar el 4 (CAPS SHIFT más 0) para insertar después un 3 en su lugar. Pulsar ENTER y la nueva línea subirá al programa sin interferir con la 4020. Este método es mucho más fácil que escribir todo otra vez, y hay menos probabilidades de cometer errores.

Las instrucciones PLOT y DRAW de las líneas 3030 y 3040 dibujan las líneas que enmarcan la sección de las fichas de dominó. Si se han introducido correctamente, RUN 3000 producirá el resultado mostrado en la figura 2.4.

La introducción del resto de la pantalla del juego es directa. Ha de incluir un título, unas instrucciones que expli-

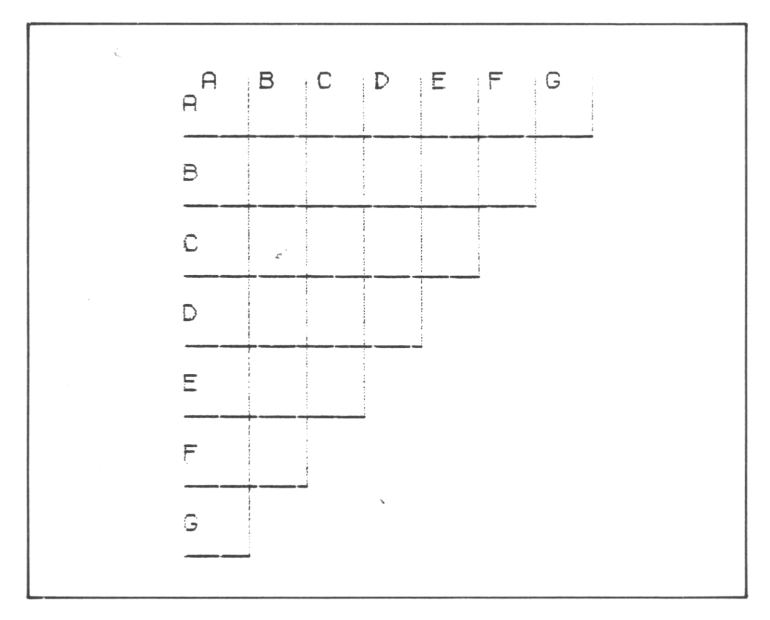


Figura 2.4

quen las reglas, el marcador y una X que indique el sitio donde va a aparecer la ficha. Las líneas 3050 y 3110 surten estos efectos. Introducir ahora:

```
3050 PRINT INVERSE 1; INK 1;AT 8,20;"SOLITA
     RIO";AT 9,20;"DE DOMINO"
3060 PRINT AT 11,14;"PULSAR S (SI)";AT 12,1
     4;"O N (NO)"
3070 PRINT AT 14,11;"INTRODUCIR LETRA IZQ";
     AT 15,11;"Y LUEGO LETRA SUP"; INK 1;AT
     16,12;"INDEPENDIENTEMENTE."
3080 PRINT INK 1;AT 0,24;"PUNTOS": PRINT AT
     2,22;"A FAVOR:";AT 4,20;"EN CONTRA:";A
     T 6,24;"MEDIA:"
3090 LET FX=0: LET CX=0: LET ME=0
3100 PRINT AT 2,30;FX;AT 4,30;CX;AT 6,30;ME
3110 PRINT AT G,C+2;"X"
```

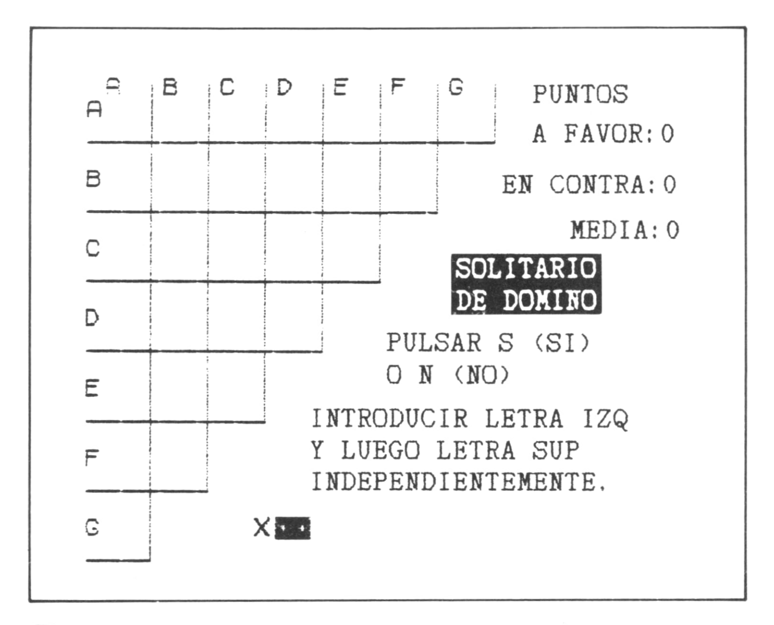


Figura 2.5

Si se ha introducido todo esto correctamente, el resultado debe ser la pantalla mostrada en la figura 2.5. En caso contrario, repasar las entradas.

Hay unos cuantos puntos a destacar en las últimas entradas. El empleo del color para distinguir el título, el marcador y la palabra INDEPENDIENTEMENTE se desprende con meridiana claridad de la sucesión de instrucciones INK 1: en la línea 3080 afecta a la primera instrucción PRINT AT, que va seguida por :PRINT AT; si se hubiera escrito aquí; AT en vez de PRINT AT, las demás entradas de esa línea serían también azules.

Las líneas 3090 y 3100 introducen el principio del sistema de marcadores y asignan letras (variables) a cada posición: FX a la posición A FAVOR, CX a la posición EN CONTRA y ME a MEDIA. Estas variables son puestas a cero por la línea 3100. Esta línea 3100 introduce una vez más la X situándola

en la posición G,C+2, es decir en 19 (G),12 (C es 10, que más 2 hacen 12). Todo esto significa que la ficha de dominó elegida al azar en X se puede poner en G,D.

La sección 3000 está ya completa y la RETURN de la línea 3998 enviará el ordenador de vuelta a la línea siguiente a GO SUB 3000, que es la 100, la primera línea del juego. Hay que poner un límite al número de fichas de dominó que está permitido utilizar en el juego; la línea 110, que mostraremos en seguida, se encarga de hacerlo. Ahora necesitamos elegir una ficha al azar para presentarla en X. La función RND es una importante función que selecciona números al azar, pero ahora es preciso tomar las medidas oportunas para que cada media ficha tenga un número distinto, una necesidad cubierta también en las siguientes líneas:

```
110 IF CX=225 THEN PRINT FLASH 1;AT 11,2;"
    PARTIDA TERMINADA": GO TO 230
120 LET A$="ABCDEFG"
130 LET B$="HIJKLMN"
140 LET Z=INT (RND*7+1)
150 LET Q=INT (RND*7+1)
160 PRINT AT G,D;A$(Z);B$(Q)
170 LET K$=A$(Z)+B$(Q)
180 INPUT "PUEDE IR? (S/N):";E$
190 IF E$="N" THEN LET CX=CX+1: PRINT AT 4
    ,30;CX: GO TO 100
200 IF E$="S" THEN LET FX=FX+1: PRINT AT 2
    ,30;FX: GO TO 240
210 IF E$<>"N" OR E$<>"S" THEN GO TO 180
220 PRINT AT 6,30;"1:";INT (CX/FX):GO TO 2
    30
230 INPUT "PULSAR R PARA OTRA PARTIDA:";P$
235 IF P$="R" THEN CLS: GO TO 60
240 INPUT "DONDE?(ESCRIBIR LET.LAT Y SUP);
    "X;Y
250 PRINT AT X,Y;K$: GO TO 260
260 PRINT AT G,D;"  ": GO TO 14
```

La línea 110 estipula que la partida llegará a su fin cuando hayamos utilizado 225 fichas en nuestros esfuerzos por completar el juego (enviando a tal efecto el ordenador a las líneas 220 y 230, que presentan el marcador de la media y preguntan si queremos jugar otra partida). Se puede alterar ese número de fichas a voluntad, pero el 225 da suficientes oportunidades de ganar. Al final de la partida, la variable ME da la media.

Las líneas 120 y 130 son una copia de las líneas 4030 y 4040 y por lo tanto se pueden grabar editando estas últimas tal como se ha explicado anteriormente. Las líneas 140 y 150 eligen al azar diferentes números para cada una de las medias fichas (que es otra razón por la que hemos utilizado en este juego dos conjuntos de medias fichas).

La PRINT AT G,D de la línea 160 muestra el empleo de las letras A a la G una vez más, y la línea 170 junta las dos medias fichas así obtenidas y las pone en la cadena K$. La pregunta INPUT de la línea 180 da comienzo a la partida. Hay que responder a esa pregunta. Si la respuesta es N, la línea 190 hace que el ordenador sume un 1 al marcador EN CONTRA, mediante LET CX=CX+1, y que imprima el resultado. Si la respuesta es S, la línea 200 suma un 1 al marcador A FAVOR mediante LET FX=FX+1, imprime el resultado y salta a la siguiente INPUT (línea 240). Esta INPUT (DONDE?) pide una posición X,Y (a expresar mediante las letras A a la G), y la línea 250 imprime en X,Y la ficha compuesta presentada anteriormente en X. El ordenador pasa después a la línea 140 para presentar una nueva ficha e imprimirla tras borrar la anterior.

Con estas líneas el juego está virtualmente completo, pero un poco de música a base de pitidos (BEEP) le daría un poco más de vida. Por eso añadimos ahora a la línea 110 un bucle FOR/NEXT, y otro igual a la línea 260.

```
110 IF CX=225 THEN PRINT FLASH 1;AT 11,2;"
    PARTIDA TERMINADA": FOR X=0 TO 20: BEE
    P .1,X: NEXT X: GO TO 230
```

```
260 PRINT AT G,D;"  ": FOR X=0 TO 20: BEEP
    .1,X: NEXT X: GO TO 140
```

Una vez hechas estas entradas, lo más lógico es que se quiera jugar una partida, pero queda por hacer todavía algo que sin ser absolutamente esencial sí da un toque de limpieza al programa: borrar todas las líneas STOP (2999, 3999 y 4999) pues ya no hacen nada.

Quien esté firmemente decidido a escribir cuanto antes sus propios programas, que repase otra vez todo el capítulo. Con una segunda lectura quedarán claros muchos puntos y además se comenzará a vislumbrar la importancia de componer una especie de estructura del programa antes de ponerse a escribir las líneas concretas. Todas las secciones son razonablemente fáciles de desarrollar una por una, pero si se hubiera enfocado la totalidad del programa sin partir de un plan bien meditado (lo que antes llamábamos esqueleto), todo habría sido mucho más difícil.

3. LANZAMIENTO DE DADOS

Mientras trabajaba en el juego del solitario con fichas de dominó se me ocurrió que media ficha de dominó constituiría un buen punto de partida para representar un dado, y luego me vino a la cabeza un antiguo juego de azar en el que se tiran tres dados. Se apuesta a un número y si sale una vez se recupera el dinero apostado, si sale dos veces se dobla esa cantidad y si sale tres veces se triplica. Sabía que haría un buen juego de ordenador, y, además, que se podría dividir fácilmente en secciones para formar un esqueleto, como hicimos en el juego anterior.

La sección gráfica que genera los dados se puede tomar prestada del programa del solitario con fichas de dominó, usando sólo la media ficha de dominó y sin la ficha blanca. Esta sección podría ir a partir de la línea 5000. Falta componer la pantalla, línea 4000 en adelante, tirar los dados, línea 3000 en adelante, convertir los dados en números normales y estructurar un marcador, línea 2000 en adelante, y determinar

las acciones del juego, línea 1000 en adelante. Todo ello da lugar al siguiente esqueleto:

```
  10 REM PRUEBE SU SUERTE
  20 BORDER 3: PAPER 6: CLS
  30 POKE 23609,255: POKE 23658,8
  40 GO SUB 5000
  50 GO SUB 4000
  60 GO SUB 3000
  70 GO SUB 2000
1000 REM JUEGO
1999 STOP
2000 REM DADOS A NUMEROS
2998 RETURN
2999 STOP
3000 REM LANZAMIENTO DE DADOS
3998 RETURN
3999 STOP
4000 REM DISENO DE LA PANTALLA
4998 RETURN
4999 STOP
5000 REM CARACTERES GRAFICOS
5998 RETURN
5999 STOP
```

Como siempre, empezaremos por la última sección, los gráficos. Para representar los dados sólo se necesitan seis caracteres, pero hace falta un carácter más ("G") que dibuje en la pantalla una mesa por la que ruedan los dados, luego la primera línea comenzará con "FOR X=0 TO 6" (7 en total). Para representar los dados se pueden utilizar las líneas DATA del juego anterior, líneas 5040 y 5050, pero habrá que descontar ocho números que correspondían a la ficha blanca y añadir ocho números adicionales y asociarlos a la "G". Introducir estas líneas:

```
5010 FOR X=0 TO 6: FOR Y=0 TO 7: READ Z: PO
     KE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT X
```

```
5020 DATA 255,255,255,247,247,255,255,255,2
     55,253,253,255,255,223,223,255,255,253
     ,253,247,247,223,223,255,255,221,221,2
     55,255,222,223,255:REM AABBCC
5030 DATA 255,221,221,255,255,221,221,255,2
     55,221,221,247,247,221,221,255,255,213
     ,213,255,255,213,213,255,85,0,85,0,85,
     0,85,0: REM DDEEFFGG
```

Si ahora ejecutamos este programa y lo listamos después, las instrucciones REM de las líneas 5020 y 5030 mostrarán los caracteres gráficos asociados a cada letra. Obsérvese que el carácter gráfico nuevo, el asociado a la G, es verde. Será verde únicamente si se ha especificado "PAPER 6" (un fondo amarillo) en la línea 20, pues si se hubiera escrito "PAPER 5" (cian), ese gráfico sería azul oscuro, y si se hubiera escrito "PAPER 7" (blanco), sería gris. Queremos que el carácter asociado a la G sea verde para que represente al tapete verde de las mesas de juego. Si alguno de los dados así representado no tuviera la forma que debería tener, repasar las instrucciones DATA.

A continuación hay que componer la pantalla. Para hacer esto, y para todo tipo de trabajos que impliquen trabajar con posiciones de la pantalla, es altamente recomendable utilizar un tablero con un revestimiento de melamina, surcado por líneas que definen cuadros correspondientes a las posiciones de la pantalla. Tiene en su lado izquierdo una columna de números PRINT AT X, y en su borde superior una fila de números PRINT AT Y. Las coordenadas PLOT/DRAW correspondientes al eje X están marcadas en el borde inferior y las correspondientes al eje Y en el lateral derecho.

Con este tablero es sumamente fácil marcar a lápiz cualquier posición de la pantalla para leer sin problemas sus coordenadas X,Y. Con una goma de borrar normal se puede borrar el tablero dejándolo listo para la próxima vez. Si se hace cada dibujo con un lápiz de punta fina de forma que ocupe

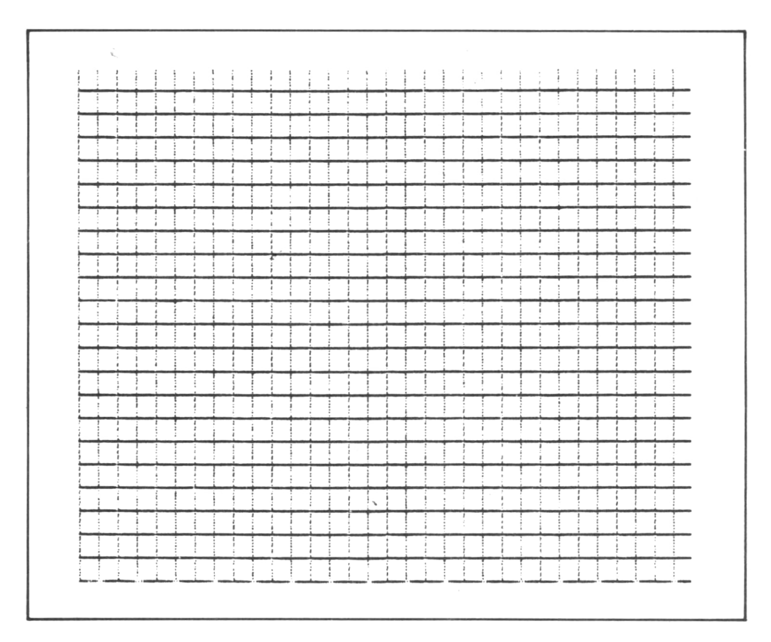


Figura 3.1

los espacios que tiene que ocupar (cada letra o carácter ocupa un cuadro), es fácil ver qué espacios quedan libres para las demás cosas que se quiera incluir. Por ejemplo: si estamos dibujando el tapete verde por el que rodarán los dados, después de dibujarlo a lápiz en el tablero se ven claramente los espacios que quedan libres para poner las diferentes partes del marcador o las instrucciones del juego.

Si no se puede conseguir este tablero, dibujar uno sobre una lámina de papel o cartón trazando a tinta las líneas necesarias para que haya 21 × 31 cuadros como éstos, figura 3.1.

Numerar en el lateral izquierdo las filas, de 0 a 21 en orden descendente, y en el borde superior las columnas, de 0 a 31 de izquierda a derecha. Numerar en el borde inferior los pixels PLOT/DRAW del eje X, de 0 a 255 en pasos de 8 (línea del

borde = 0, primera línea = 8, segunda línea = 16, etc.), y en el lateral derecho los pixels PLOT/DRAW del eje Y, de 0 a 175 en saltos de 8 y en orden ascendente (línea de borde = 0, primera línea = 8, etc.). Si el tablero es de $9\frac{1}{2} \times 6\frac{1}{2}$ pulgadas, será de un tamaño parecido a la pantalla. El tablero así construido debe servir para hacer muchas prácticas si se utiliza un lápiz y se borra con cuidado.

Olvidémonos del método concreto utilizado para componer la pantalla y pasemos ahora a los detalles de la sección 4000, dedicada a programar la pantalla. Primero va el título, seguido por el tapete de mesa, las diversas posiciones del marcador y algo de información. La pantalla compuesta para este programa es la del listado siguiente, líneas 4010 a 4060.

```
4010 PRINT INVERSE 1; INK 1;AT 1,6;"* * PRU
     EBE SU SUERTE * "
4020 PRINT AT 3,1;"APUESTE A UN NUMERO (1-6
     )";AT 5,16;"TIENE PTS:"
4030 FOR X=6 TO 12: FOR Y=0 TO 13: PRINT AT
     X,Y;"G";AT X,14;"8": NEXT Y: NEXT X
4040 FOR Y=0 TO 14: PRINT AT 5,Y;"8";;AT 13
     ,Y;"8": NEXT Y
4050 PRINT AT 15,1;"1 ACIERTO: DINERO ATRAS
     ";AT 16,1;"2 ACIERTOS: GANA EL DOBLE";
     AT 17,1;"3 ACIERTOS: GANA EL TRIPLE"
4060 PRINT AT 7,18;"MARCADOR:";Z: LET Z=0:
     PRINT AT 9,16;"DADO 1:";AT  10,16;"DAD
     O 2:";AT 11,16;"DADO 3:"
```

Una vez introducidas y revisadas estas líneas, ejecutarlas ignorando el mensaje del pie de la pantalla (7 RETURN WITHOUT GO SUB) y se verá un tapete verde sobre fondo negro y las leyendas. Obsérvese que la instrucción "LET Z=0" puesta en la línea 4060 inmediatamente después de la entrada "MARCADOR", hace que salga un cero a continuación de MARCADOR. Esa Z es una "variable" que se utilizará muy a menudo más adelante.

Los cuadros negros de las líneas 4030 y 4040 son los caracteres correspondientes a la pulsación de las teclas CAPS SHIFT y 8 cuando se está en el modo gráfico. El cuadrado de trazos de la línea 4030 es el gráfico asociado a la tecla "G".

Como la sección 4000 ya está completa, pasemos a la sección 3000, "Lanzamiento de los dados". Merece la pena dedicar a esta sección unos momentos de reflexión con objeto de intentar dar la impresión de que los dados ruedan por la mesa hasta su posición final. Además, queremos que la cara o número de cada dado sea diferente de las de los otros dos, al azar.

Para dar la ilusión de movimiento hay que imprimir el objeto móvil en una posición, borrarlo, imprimirlo en la posición siguiente, borrarlo, imprimirlo en la posición siguiente, y así sucesivamente. Si queremos que el movimiento quede representado sobre un tablero verde, habrá que sustituir cada vez el dado por el gráfico asociado a la "G", para lo cual se necesita otro bucle FOR/NEXT. Introducir el listado siguiente y después haremos una prueba con un solo dado.

```
3010 LET A$="ABCDEF"
3020 RANDOMIZE 0
3030 LET L=INT (RND*6+1)
3040 LET M=INT (RND*6+1)
3050 LET N=INT (RND*6+1)
3060 LET X=7: FOR Y=0 TO 9
3070 PRINT AT X,Y;A$(L): PAUSE 2: PRINT AT
     X,Y;"G": NEXT Y: PRINT AT 7,10;A$(L):
     BEEP .25,30
```

Si se han introducido estas líneas correctamente, RUN presentará la pantalla de las líneas 4010 a 4060 y por ella rodará un dado. Si se quiere ver otra vez, introducir RUN 3000 y ENTER. El dado rodará otra vez sin alterar la pantalla. La impresión final del dado se produce a la conclusión del bucle FOR/NEXT, y va acompañada por un pitido (BEEP).

Ahora hay que sacar a escena los otros dos dados. Obsérvese que en el primero se ponía una L tras la A$. En los otros dos pondremos M y N tras sus A$ para garantizar que normalmente salen resultados distintos. Habrá aciertos dobles y hasta algún triple, pero a fin de cuentas se acabará perdiendo dinero. Se ha incluido la instrucción "PAUSE 2" para hacer más pronunciada la ilusión de movimiento; sin esa pausa todo ocurriría demasiado rápidamente y con una pausa más larga la acción sería demasiado obvia. Hacer pruebas alterando ese número para ver los efectos que surte; será altamente instructivo.

Los otros dos dados se pararán en sitios distintos con objeto de que la acción no parezca demasiado mecánica. Introducir el resto de esta sección.

```
3080 LET X=9: FOR Y=0 TO 11
3090 PRINT AT X,Y:A$(M): PAUSE 2: PRINT AT
     X,Y;"G": NEXT Y: PRINT AT 9,12;A$(M):
     BEEP .25,25
3100 LET X=11: FOR Y=0 TO 10
3110 PRINT AT X,Y;A$(N): PAUSE 2: PRINT AT
     X,Y;"G": NEXT Y: PRINT AT 11,11;A$(N):
     BEEP .25,30
```

Como ya está concluida la sección 3000, pasamos a la 2000, que tiene que cambiar los dados que hay encima del tapete por números normales a presentar en el marcador. Esta sección es muy repetitiva, pero éste es el método más fácil y más al alcance de los principiantes. Se necesita la instrucción IF/THEN, una de las que el ZX Spectrum sabe obedecer con mayor prontitud y eficacia.

En primer lugar hay que repartir algunas letras (llamadas variables) por las posiciones siguientes a las que ocupan los dados 1, 2 y 3. Son las letras Q, R y S. Procedamos: IF A$(L) (el dado que está en la posición más alta)=(es igual a) A$(1) (la primera porción de la cadena A$) THEN LET Q=1:

PRINT AT 9,23; Q (la posición de Q es la siguiente al dado 1). Hay que poner una de estas líneas para cada uno de los seis números o caras del dado, en cada una de las tres posiciones que ocupan los dados, lo que hace 18 líneas en total. Introducir las líneas 2010 a 2180 para completar la sección.

```
2010 IF A$(L)=A$(1) THEN LET Q=1: PRINT AT
     9,23;Q
2020 IF A$(L)=A$(2) THEN LET Q=2: PRINT AT
     9,23;Q
2030 IF A$(L)=A$(3) THEN LET Q=3: PRINT AT
     9,23;Q
2040 IF A$(L)=A$(4) THEN LET Q=4: PRINT AT
     9,23;Q
2050 IF A$(L)=A$(5) THEN LET Q=5: PRINT AT
     9,23;Q
2060 IF A$(L)=A$(6) THEN LET Q=6: PRINT AT
     9,23;Q
2070 IF A$(M)=A$(1) THEN LET R=1: PRINT AT
     10,23;R
2080 IF A$(M)=A$(2) THEN LET R=2: PRINT AT
     10,23;R
2090 IF A$(M)=A$(3) THEN LET R=3: PRINT AT
     10,23;R
2100 IF A$(M)=A$(4) THEN LET R=4: PRINT AT
     10,23;R
2110 IF A$(M)=A$(5) THEN LET R=5: PRINT AT
     10,23;R
2120 IF A$(M)=A$(6) THEN LET R=6: PRINT AT
     10,23;R
2130 IF A$(N)=A$(1) THEN LET S=1: PRINT AT
     11,23;S
2140 IF A$(N)=A$(2) THEN LET S=2: PRINT AT
     11,23;S
2150 IF A$(N)=A$(3) THEN LET S=3: PRINT AT
     11,23;S
2160 IF A$(N)=A$(4) THEN LET S=4: PRINT AT
     11,23;S
```

```
2170 IF A$(N)=A$(5) THEN LET S=5: PRINT AT
     11,23;S
2180 IF A$(N)=A$(6) THEN LET S=6: PRINT AT
     11,23;S
```

Una forma fácil de introducir repeticiones como ésta consiste en escribir totalmente la primera línea y, una vez que ha subido al programa, volverla a llamar (con CAPS SHIFT y 1) y luego EDITarla en la línea siguiente con la ayuda del cursor. Antes de nada, y en todos los casos, cambiar el número de línea pues es ciertamente desesperante encontrarse con que al pulsar ENTER después de haber cambiado toda la línea, la nueva línea va a sustituir a la línea anterior simplemente porque se ha olvidado cambiar el número de línea.

Después de cambiar el número de línea, cambiar el 1 por un 2 y Q=1 por Q=2. Repetir estos cambios en las seis primeras líneas, y cuando se llegue a las siguientes no olvidar que hay un cambio brusco de la serie Q por la serie R y por la serie S. Son pequeñas sorpresas que alivian un poco el tedio de tanta repetición mecánica.

Una vez introducidas correctamente todas las líneas, ejecutar el programa y ver si en las posiciones del MARCADOR salen los números correspondientes a las caras de los dados. Con esto queda completada la sección 2000 y está expedito el camino que lleva a desarrollar las acciones reales del juego, sección 1000. Como esta sección tiene que incluir muchos controles para llevar las cuentas de los marcadores alternativos, tendrá que ser más complicada.

Hay muchas “variables” por conocer:

Z es el MARCADOR e irá a parar a la posición 7,27;
Q es el DADO 1 e irá a parar a la posición 9,23;
R es el DADO 2 e irá a parar a la posición 10,23;
S es el DADO 3 e irá a parar a la posición 11,23;
JX es el dinero disponible e irá a parar a la posición 5,26;

A es la cantidad restada o sumada a JX;
B es el número al que se apuesta e irá a parar a 19,21.

Las tres últimas, JX, A y B, están por definir. La cantidad de dinero que podemos usar para hacer apuestas (JX) será fijada a 50 ptas. y será impresa en 5,26; lo hará la línea 1010. Luego se necesita una INPUT que permita introducir la apuesta, que será asignada a la variable A; lo hará la línea 1020. La línea 1040 se encargará de restar de la cantidad disponible (JX) la cantidad apostada (A), mediante la instrucción "LET JX=JX–A", y de imprimir luego el valor de JX en 5,26, tras borrar el 50 que había.

Tiene que haber otra INPUT que permita seleccionar el número (1 al 6) al que se apuesta, que se asigna a la variable B; está en la línea 1050. La línea 1060 se encarga de excluir los números distintos al 1 al 6, y la línea 1070 imprime B en 19,21.

Ahora hay que tirar los dados, para lo cual se incluye una GO SUB 3000 seguida por una GO SUB 2000, borrando por lo tanto las líneas 60 y 70 anteriores. No olvidar borrar esas líneas. Esta última acción muestra las ventajas de escribir las distintas partes del programa en diferentes secciones, un principio que hay que observar siempre. Estas líneas provocan el lanzamiento de los dados y la impresión de los números correspondientes en DADO 1, DADO 2 y DADO 3.

En la sección 2000 asignábamos a Q, R y S la función de representar a los números o caras de los dados: Q correspondía al dado 1, R al dado 2 y S al 3. Ahora, en las líneas 1100 a 1120, comparamos B (el número apostado) con esos números, uno por uno, para ver si coincide con alguno de ellos. En caso positivo incrementamos Z (el marcador) en 1 e imprimimos un * tras el número para que llame la atención. Luego imprimimos el nuevo valor de Z (línea 1130).

Si Z (el marcador) sigue siendo 0 imprimimos un mensaje "PIERDE" (línea 1140) y enviamos el ordenador a la línea

1500. Si Z es 1 imprimimos un mensaje “GANA” (línea 1150) y sumamos el dinero ganado al dinero disponible (JX). Las líneas 1160 y 1170 hacen lo mismo para el caso de que Z sea 2 o sea 3, respectivamente.

La línea 1500 examina JX (el dinero disponible) y si es 0 (si se ha gastado todo) envía el control del programa a la línea 1900. Si todavía hay dinero disponible envía el control del programa a las líneas 1510 y 1520 para que suene un pitido (BEEP) y ofrezca otra jugada. Si la respuesta a este ofrecimiento es N (NO), la línea 1530 envía el control del programa a la línea 1900 y si es S (SI) borra los dados de la mesa, borra también los marcadores y envía el control del programa a la línea 1020 para que dé comienzo otra secuencia.

La línea 1900 borra la pantalla, imprime el mensaje “GRACIAS” y ofrece otra partida. Repasar una y otra vez estas últimas líneas hasta tener la total seguridad de que se entiende perfectamente cómo y por qué pasa lo que pasa. Introducir ahora estas líneas:

```
1010 LET JX=50: PRINT AT 5,26;JX
1020 INPUT "HAGA SU APUESTA:";A
1040 LET JX=JX-A: PRINT AT 5,26;"  ": PRINT
     AT 5,26;JX: PRINT AT 19,1;"APUESTA ";A
     ;" PTS AL"
1050 INPUT "A QUE NUMERO (1-6):";B
1060 IF B<1 OR B>6 THEN GO TO 1050
1070 PRINT AT 19,23;B: PAUSE 10
1080 GO SUB 3000
1090 GO SUB 2000
1100 IF Q=B THEN LET Z=Z+1: PRINT AT 9,25;"
     *"
1110 IF R=B THEN LET Z=Z+1: PRINT AT 10,25:
     "*"
1120 IF S=B THEN LET Z=Z+1: PRINT AT 11,25:
     "*"
1130 PRINT AT 7,27;"  ";AT 7,27 Z
```

```
1140 IF Z=0 THEN PRINT AT 19,1;"PIERDE
                    ": GO TO 1500
1150 IF Z=1 THEN PRINT AT 19,1;"GANA
                  ": LET JX=JX+A: PRINT AT 5
     ,26;"   ": PRINT AT 5,26;JX: GO TO 15
     00
1160 IF Z=2 THEN PRINT AT 19,1;"GANA
                  ": LET JX=JX+(A*2): PRINT
     AT 5,26;"   ": PRINT AT 5,26;JX: GO TO
     1500
1170 IF Z=3 THEN PRINT AT 19,1;"GANA
                  ": LET JX=JX+(A*3): PRINT
     AT 5,26;"   ": PRINT AT 5,26;JX: GO TO
     1500
1500 IF JX=0 THEN PRINT AT 19,1;"NO LE QUE
     DA DINERO": PAUSE 50: GO TO 1900
1510 FOR N=1 TO 3: BEEP .25,25: BEEP.25,35:
     NEXT N
1520 INPUT FLASH 1;"OTRA JUGADA? (S/N)";K$
1530 IF K$="N" THEN GO TO 1900
1540 IF K$="S" THEN GO TO 1550
1550 LET Z=0: PRINT AT 7,27;Z;AT 19,1;"
                         "
1560 PRINT AT 7,10;"G";AT 9,12;"G";AT 11,11
     ;"G";AT 9,23;"   ";AT 10,23;"   ";AT 1
     1,23;"   ": GO TO 1020
1900 CLS: PRINT AT 10,5;"GRACIAS POR JUGAR"
     : FOR X=1 TO 26: BEEP .1,X: NEXT X
1910 PRINT AT 12,3;"PULSE RUN PARA OTRA PAR
     TIDA"
```

Prestar la máxima atención al introducir las líneas. Si el programa no funciona bien cuando se intenta ejecutar después, revisar las líneas una por una contrastándolas con el listado del libro porque éste es el listado impreso directamente por la impresora del ordenador después de haber ejecutado con éxito el programa.

4. EL TABLERO Y JUEGO DE DAMAS

Después de los juegos con dados y con fichas de dominó, para variar decidimos hacer un juego sencillo: un juego de damas para dos jugadores. Es posible escribir juegos en los que participe el ordenador, pero pueden resultar muy complicados y el objetivo de este libro es ayudar a los principiantes a escribir sus programas de la manera más fácil posible; por esa razón incluimos sólo ejemplos sencillos.

Una vez más, levantaremos en primer lugar un esqueleto del juego constituido por una sección dedicada a los caracteres gráficos, otra para la composición de la pantalla y las reglas, otra para los movimientos de las negras, y otra para los movimientos de las blancas. Los gráficos estarán en la sección 5000, la pantalla en la 4000, los movimientos de las negras en la 1000 y los movimientos de las blancas en la 2000. La primera entrada es por lo tanto:

```
10 REM DAMAS .
20 BORDER 4: PAPER 7: CLS
30 POKE 23658,8: POKE 23609,255
40 GO SUB 5000
50 GO SUB 4000
```

```
1000 REM MOVIMIENTOS DE NEGRAS
1999 STOP
2000 REM MOVIMIENTOS DE BLANCAS
2999 STOP
4000 REM DIBUJO DE LA PANTALLA
4999 STOP
5000 REM CARACTERES GRAFICOS
5999 STOP
```

La mayoría de las secciones son bastante cortas pero de todas formas están bien delimitadas. La sección de los caracteres gráficos es la que abordaremos en primer lugar. Hay cuatro caracteres gráficos: dos piezas, una negra y otra blanca, y dos coronas, una blanca y otra negra. El lector debe estar a estas alturas suficientemente capacitado para diseñarlos sin ayuda, pero daremos las líneas DATA de los que hemos diseñado.

```
5010 FOR X=0 TO 3: FOR Y=0 TO 7: READ Z: PO
     KE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
5020 DATA 60,66,129,129,129,129,66,60,60,12
     6,255,255,255,255,126,60:REM AABB
5030 DATA 165,255,129,129,129,129,66,60,165
     ,255,255,255,255,255,126,60: REM CCDD
5100 RETURN
```

Esto completa la sección de los caracteres gráficos y ahora viene la sección de la pantalla. Tendremos que reflexionar mucho más si queremos poner en la pantalla un tablero de tamaño razonable. Muchos juegos de damas tienen tableros que son demasiado pequeños. Un tablero con cuadros de cuatro caracteres (2 × 2) y piezas blancas y negras del tamaño de un carácter es más o menos adecuado. Para ilustrar lo que queremos decir introducir el siguiente listado. Está suficientemente alejado del programa por lo que no hay peligro de que interfiera.

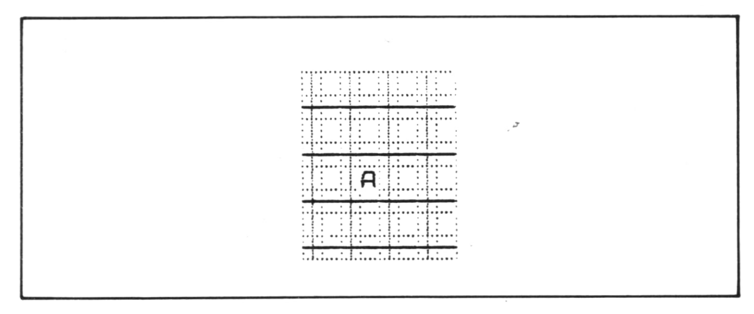
```
9000 REM PRUEBA
9010 FOR A=0 TO 64 STEP 2: FOR X=0 TO 64 ST
     EP 8: PLOT 48+A,80+X: NEXT X: NEXT A
9020 FOR A=0 TO 64 STEP 2: FOR X=0 TO 64 ST
     EP 8: PLOT 48+X,80+A: NEXT X: NEXT A
```

Estas líneas dibujan en la pantalla una cuadrícula de líneas de puntos que indican las posiciones PRINT AT. Como queremos que una pieza ocupe uno de esos cuadros y que a la vez quede en el centro de un cuadro más grande, trazamos las líneas que queremos que marquen las divisiones del tablero. Vemos que para cumplir esta condición esas líneas tienen que pasar por la mitad de algunas posiciones PRINT AT. Introducir las líneas siguientes:

```
9030 FOR X=0 TO 48 STEP 16: PLOT 48,X+84: D
     RAW 64,0: NEXT X
9040 FOR X=0 TO 48 STEP 16: PLOT 52+X,80: D
     RAW 0,64: NEXT X
9050 PRINT AT 8,9;"A"
```

Se ha puesto una línea extra que imprima una pieza en un cuadro (la A introducida en el modo gráfico). Si se ejecutan estas líneas con RUN 9000 se tendrá el resultado mostrado en la figura 4.1, que es justo lo que queríamos.

Figura 4.1



Antes de seguir adelante, intentar descubrir cómo se las arreglan estas pocas líneas (9000 a 9050) para hacer lo que hacen. Las líneas de puntos tienen un interés especial. Van saliendo lentamente pues son en realidad una serie de puntos impresos por instrucciones PLOT, no líneas interrumpidas. Para hacerlas se ha utilizado un bucle X anidado en el interior de un bucle A; la única diferencia entre las líneas 9010 y 9020 está en la parte PLOT de la instrucción: PLOT 48+A,80+X en una y PLOT 48+X,80+A en la otra. El bucle A imprime un punto en posiciones alternas (STEP 2) extendidas a lo largo de la línea de 64 pixels (FOR A=0 TO 64 STEP 2), y el bucle X marca una separación de 8 pixels entre los puntos iniciales de las líneas de pixels alternos. Cuando A está en la primera parte de la instrucción PLOT, (A,), los puntos definen líneas horizontales, y cuando A está en la segunda parte, (,A), las líneas de puntos son verticales. Hacer una copia de las líneas 9000 a 9050 para uso futuro. Puede ser útil.

Con la ayuda de este pequeño programa podemos dibujar el tablero. Hemos dibujado en el tablero de melamina la forma y posición del tablero de ajedrez y el resultado de ello son las líneas 4010 en adelante. Introducirlas.

```
4010 FOR X=0 TO 128 STEP 16: PLOT 2020,X+36
     : DRAW 128,0: NEXT X
4020 FOR X=0 TO 128 STEP 16: PLOT X+20,36:
     DRAW 0,128: NEXT X
4030 PRINT INVERSE 1; INK 1;AT 0,2;"* D A M
     A S * *"
4040 PRINT AT 18,3;"A B C D E F G H";AT 16,
     0;16;AT 14,0;14;AT 12,0;12;AT 10,0;10;
     AT 8,1;8;AT 6,1;6;AT 4,1;4;AT 2,1;2
4050 LET A=3: LET B=5: LET C=7: LET D=9: LE
     T E=11: LET F=13: LET G=15: LET H=17
```

Una vez introducidas, ejecutar el programa. Debe presentar en pantalla un tablero grande con números a lo largo del lateral izquierdo y letras a lo largo del borde inferior. En caso

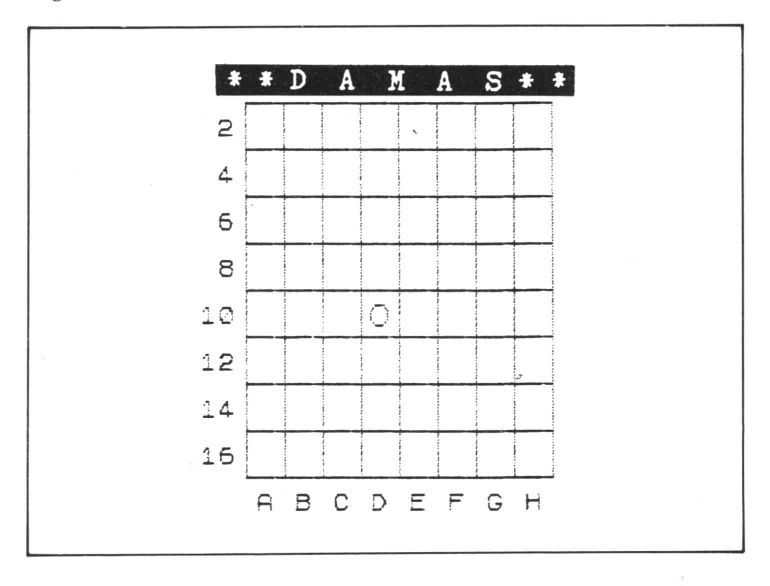
contrario, repasar las entradas porque el listado del libro es correcto. Se verá que los números de la izquierda y las letras del pie están centrados con respecto a los cuadros. Para ver la posición de la pieza en el tablero añadir una línea provisional con el número 4500 (pulsar la "A" en el modo gráfico).

```
4500 PRINT AT 10,D;"A"
```

Cuando se ejecute el programa saldrá una pieza blanca situada en el centro de uno de los cuadros (ver figura 4.2). Con esta línea se comprueban las posiciones de la pantalla y la validez de la definición de las posiciones PRINT AT mediante un número y una letra.

Como todo parece ir bien, podemos borrar la línea 4500. Para hacerlo basta escribir 4500 y pulsar ENTER. La línea desaparecerá. No olvidar lo que acaba de pasar: si alguna vez se escribe accidentalmente un número de línea y luego se

Figura 4.2



pulsa ENTER, se habrá borrado esa línea. Un error muy normal es escribir 2500 en vez de 3500 y borrar sin darse cuenta la línea 2500 escrita antes. Muchas veces no se sabe por qué no funciona un programa y al repasar las líneas se descubre que falta alguna línea que no se recuerda haber borrado. Como esto puede ser realmente molesto, al introducir líneas de programa escribir la primera orden de la línea nada más escribir el número de línea, en todo caso antes de ir a buscar al listado la siguiente parte de esa línea. La razón está en que las líneas escritas a medias no pueden entrar al ordenador hasta que no estén completas. (Hacer la prueba con 2500 IF y ENTER.)

El tablero de damas está dividido en cuadros blancos y negros (es igual que el de ajedrez), luego lo primero que hay que hacer es “pintar” los cuadros negros. El problema es que no podemos utilizar la instrucción PRINT AT con gráficos negros, que sería la solución más rápida, porque los espacios PRINT AT están cortados por las líneas del tablero. Es preciso recurrir por lo tanto a la orden PLOT DRAW para pintar los 32 cuadros negros, y también aquí se vuelve a poner de manifiesto la necesidad de un tablero de dibujo donde dibujar las posiciones de la pantalla. Hay que determinar cada posición PLOT con total exactitud sirviéndose a tal efecto de la fila de números de pixels del pie del tablero para conocer la primera coordenada (eje X), y de la columna de números de pixels del lateral derecho para conocer la segunda coordenada (eje Y). Las líneas 4060 a 4130 lo hacen.

```
4060 FOR X=1 TO 15: PLOT 20,36+X: DRAW 16,0
     : PLOT 52,36+X: DRAW 16,0: PLOT 84,36+
     X: DRAW 16,0: PLOT 116,36+X: DRAW 16,0
     : NEXT X
4070 FOR X=17 TO 31: PLOT 36,36+X: DRAW 16,
     0: PLOT 68,36+X: DRAW 16,0: PLOT 100,3
     6+X: DRAW 16,0: PLOT 132,36+X: DRAW 16
     ,0: NEXT X
4080 FOR X=33 TO 47: PLOT 20,36+X: DRAW 16,
```

```
     0: PLOT 52,36+X: DRAW 16,0: PLOT 84,36
     +X: DRAW 16,0: PLOT 116,36+X: DRAW 16,
     0: NEXT X
4090 FOR X=49 TO 63: PLOT 36,36+X: DRAW 16,
     0: PLOT 68,36+X: DRAW 1,0: PLOT 100,36
     +X: DRAW 16,0: PLOT 132,36+X: DRAW 16,
     0: NEXT X
4100 FOR X=65 TO 79: PLOT 20,36+A: DRAW 16,
     0: PLOT 52,36+X: DRAW 16,0: PLOT 84,36
     +X: DRAW 16,0: PLOT 116,36+X: DRAW 16,
     0: NEXT X
4110 FOR X=81 TO 95: PLOT 36,36+X: DRAW 16,
     0: PLOT 68,36+X: DRAW 16,0: PLOT 100,3
     6+X: DRAW 16,0: PLOT 132,36+X: DRAW 16
     ,0:NEXT X
4120 FOR X=97 TO 111: PLOT 20,36+X: DRAW 16
     ,0: PLOT 52,36+X: DRAW 16,0: PLOT 84,3
     6+X: DRAW 16,0: PLOT 116,36+X: DRAW 16
     ,0: NEXT X
4130 FOR X=113 TO 127: PLOT 36,36+X:DRAW 16
     ,0: PLOT 68,36+X: DRAW 16,0: PLOT 100,
     36+X: DRAW 16,0: PLOT 132,36+X: DRAW 1
     6,0: NEXT X
```

Copiar todas estas líneas es indudablemente un trabajo tedioso que además exige tener mucho cuidado con los números y signos de puntuación. Es fácil equivocarse, pero cuando se consigue introducirlas correctamente se tiene la satisfacción de ver cómo "pintan" los 32 cuadros. Falta poner las piezas en el tablero, lo que se hace con dos líneas de bucles FOR/NEXT. Obsérvese que las piezas blancas están asociadas a la "A" y las piezas negras a la "B", siempre en el modo gráfico. He aquí las dos líneas en cuestión.

```
4140 FOR Y=3 TO 17 STEP 4: PRINT AT 2,Y:"A"
     ;AT 6,Y;"A";AT 14,Y;"B": NEXT Y
4150 FOR Y=5 TO 19 STEP 4: PRINT AT 4,Y;"A"
     ;AT 12,Y;"B";AT 16,Y;"B": NEXT Y
```

Si se ejecuta ahora el programa se verá en la pantalla un tablero de damas junto con sus piezas. Ha llegado el momento de añadir algunas instrucciones que expliquen los movimientos.

```
4160 PRINT AT 2,20;"PARA QUITAR";AT 3,21;"P
     IEZAS";AT 4,21;"INTRODUCIR";AT 5,21;"N
     O.LATERAL";AT 6,21;"Y LETRA.";AT 7,21;
     "SEPARADOS";AT 8,21;"----------"
4170 PRINT AT 10,20;"PARA PONER";AT 11,21;"
     PIEZAS";AT 12,21;"INTRODUCIR";AT 13,21
     ;"NO.LATERAL";AT 14,21;"Y LETRA Y";AT
     15,21;"COLOR(B/N)";AT 16,21;"SEPARADOS
     ";AT 17,21;"----------"
4180 PRINT INVERSE 1; INK 1;AT 19,2;"SIGA L
     AS INPUT ESCRUPULOSAMENTE"
4200 RETURN
```

Habiendo completado ya la sección de la pantalla, podemos pasar a los movimientos comenzando por las negras, sección 1000. La forma más sencilla de hacerlo es introducir la dirección de la pieza negra que se quiere mover, borrar esa pieza, introducir la dirección de destino e imprimir la pieza ahí. Lo hacen las ocho líneas siguientes, que además dan entrada a una rutina de “coronación”.

```
1010 INPUT "NEGRA DE? (NO. Y DESPUES LETRA)
     ";X;Y
1020 PRINT AT X,Y;" "
1030 INPUT "NEGRA A? (NO. Y DESPUES LETRA)"
     ;U;V
1040 INPUT "ES UNA CORONA? (S/N):";C$
1050 IF C$="N" THEN GO TO 1070
1060 IF C$="S" THEN GO TO 1080
1070 PRINT AT U,V;"B": GO TO 1090
1080 PRINT AT U,V;"D": GO TO 1090
```

Ahora podemos probar esta sección. A tal efecto, mover una pieza negra observando las indicaciones INPUT y siguiendo las instrucciones. Cada movimiento exige entradas independientes. Concretamente, NEGRA DE? pide lo siguiente: N.º lateral (¿12?); ENTER; letra inferior (¿B?); ENTER. La pieza desaparecerá. NEGRA A? pide algo similar: N.º, ENTER, letra y ENTER. Tras el examen para ver si es una corona o no, saldrá esa pieza en la posición designada. Con este sistema es fácil hacer trampas, pero con ello no se gana nada.

Para legislar los SALTOS, algo esencial en el juego de damas, la mejor solución, o por lo menos la más fácil, es hacer la pregunta tras cada movimiento, porque enseñar al ordenador a reconocer los saltos es un asunto muy complicado que es mejor dejar para cuando se sea un verdadero experto.

Por lo tanto, a partir de la línea 1080 hay que añadir una serie de líneas que garanticen el cumplimiento de todas las contingencias del salto, incluyendo los triples y cuádruples saltos si es necesario. La estructura es simple y fácil de seguir. La línea 1230 envía el ordenador a la línea 1120 y en el supuesto de que la respuesta a la pregunta "¿Puede dar otro salto?" sea "S" (SI), el ordenador continúa ejecutando el bucle. La respuesta "N" (NO) produce la salida del bucle y da a las piezas blancas la oportunidad de realizar un movimiento.

```
1090 INPUT FLASH 1;"HA DADO UN SALTO? (S/N)
     :";V$
1100 IF V$="N" THEN GO TO 2000
1110 IF V$="S" THEN GO TO 1120
1120 INPUT "BLANCA DE?:";X;Y
1130 PRINT AT X,Y;" ": GO TO 1140
1140 INPUT FLASH 1:"PUEDE DAR OTRO SALTO? (
     S/N):";H$
1150 IF H$="N" THEN GO TO 2000
1160 IF H$="S" THEN GO TO 1170
1170 INPUT "NEGRA DE? (NO. Y LETRA)";X;Y
```

```
1180 PRINT AT X,Y;" "
1190 INPUT "NEGRA A? (NO. Y LETRA)";U;V
1200 INPUT "ES UNA CORONA?: (S/N):";C$
1210 IF C$="N" THEN GO TO 1230
1220 IF C$="S" THEN GO TO 1240
1230 PRINT AT U,V;"B": GO TO 1120
1240 PRINT AT U,V;"D": GO TO 1120
```

Habiendo completado ya la sección 1000 que programa los movimientos de las negras, ha llegado el momento de hacer un último esfuerzo e introducir la sección 2000, los movimientos de las blancas. Todas las líneas de esta sección son una copia de las de la sección 1000, con la diferencia de que hay que cambiar la palabra NEGRA por BLANCA (en un caso hay que cambiar la palabra BLANCA por NEGRA) y comenzar todos los números de línea por 2 en vez del 1. Hay más cambios de números: cuando un número GOTO comienza por 1 hay que cambiarlo por 2, y cuando comienza por 2 hay que cambiarlo por 1. Puede ser más fácil escribir todo de nuevo que bajar las líneas anteriores una por una y cambiarlas. Que cada uno lo haga como quiera. He aquí las líneas que faltan:

```
2010 INPUT "BLANCA DE? (NO .Y DESPUES LETRA
     )";X;Y
2020 PRINT AT X,Y;" "
2030 INPUT "BLANCA A? (NO. Y DESPUES LETRA)
     ";U;V
2040 INPUT "ES UNA CORONA?: (S/N):"C$
2050 IF C$="N" THEN GO TO 2070
2060 IF C$="S" THEN GO TO 2080
2070 PRINT AT U,V;"A": GO TO 2090
2080 PRINT AT U,V;"C": GO TO 2090
2090 INPUT FLASH 1;"HA DADO UN SALTO? (S/N)
     :";V$
2100 IF V$="N" THEN GO TO 1000
2110 IF V$="S" THEN GO TO 2120
```

```
2120 INPUT "NEGRA DE?:";X;Y
2130 PRINT AT X,Y;" ": GO TO 2140
2140 INPUT FLASH 1:"PUEDE DAR OTRO SALTO? (
     S/N):";H$
2150 IF H$="N" THEN GO TO 1000
2160 IF H$="S" THEN GO TO 2170
2170 INPUT "BLANCA DE? (NO. Y LETRA)";X;Y
2180 PRINT AT X,Y;" "
2190 INPUT "BLANCA A? (NO. Y LETRA)";U;V
2200 INPUT "ES UNA CORONA?: (S/N):";C$
2210 IF C$="N" THEN GO TO 2230
1220 IF C$="S" THEN GO TO 2240
1230 PRINT AT U,V;"A": GO TO 2120
2240 PRINT AT U,V;"C": GO TO 2120
```

El método utilizado en este programa para regir los movimientos puede no ser mejor que si se estuviera jugando con piezas de verdad en un tablero de verdad, pero tampoco es peor. Por lo menos tiene la ventaja de que las piezas no se pueden mover accidentalmente ni se pueden volver a mover una vez hecho un movimiento. Hacer los movimientos sólo tras haber estudiado detenidamente los mensajes INPUT porque es muy fácil pasar por alto alguno. Si una pieza aterriza en un cuadro equivocado, la culpa es de un error cometido al hacer la entrada.

Este método ofrece sin embargo una ventaja importante: permite jugar los finales de partida que se quiera. Al ejecutar el programa, en la pantalla sale una composición normal de tablero y piezas, pero incluyendo REM en las líneas 4140 y 4150 no saldrá ninguna pieza en el tablero. Hacerlo escribiendo LIST 4140, bajando la línea para EDITarla, añadiendo REM justo a la derecha del número de línea y devolviendo arriba la línea con ENTER. Hacer lo mismo con la línea 4150.

Al ejecutar el programa sale en la pantalla el tablero pero sin fichas, y debajo sale una INPUT que pide mover las negras. Escribir STOP (SYMBOL SHIFT y A) y pulsar EN-

TER. Ahora hay en el pie de la pantalla un aviso STOP, y ése es el modo en el que se puede introducir casi cualquier cosa sin afectar a la pantalla. Para ello basta introducir PRINT AT sin ningún número de línea. He aquí un ejemplo que muestra cómo se hace.

```
PRINT AT 16,B;"C";AT 12,D;"D";AT 16,F;"C";A
      T 8,D;"D"
```

Las letras “C” y “D” se introducen en el modo gráfico. La línea que se acaba de introducir ha desaparecido ya, por lo que conviene escribirla en papel antes de pulsar ENTER. (Este método puede servir también para borrar las piezas movidas equivocadamente durante el desarrollo de una partida. Basta introducir PRINT AT?,?;“espacio”, y quedará borrada la entrada mal hecha.) Hacièndo las nuevas entradas que se quiera se puede jugar el final de partida que se quiera. Introducir ahora GO TO 1010 para mover las blancas o 2010 para mover las negras; en el pie de la pantalla saldrá la INPUT.

Una vez estudiados estos movimientos, es fácil volver a poner el ordenador en el modo original EDITando otra vez las líneas 4140 y 4150 y borrando las REM introducidas antes. Esta es una de las prestaciones que hace al programa tan práctico. Se pueden hacer muchas más adaptaciones, que se irán descubriendo según se vayan dominando los secretos de la programación. El hecho de tener estructurado el programa en secciones perfectamente aisladas y bien etiquetadas implica la ventaja de que es muy fácil moverse por él.

Queda algo por hacer: borrar las líneas STOP números 1999, 2999, 4999 y 5999. Ya no hacen ninguna función.

5. REVERSI CON GRAFICOS COMPUESTOS

El juego de Reversi nos lleva también a los métodos utilizados en los juegos anteriores pues se divide fácilmente en secciones: gráficos, pantalla, movimientos de las negras y movimientos de las blancas. El esqueleto o plan será por lo tanto similar al de los juegos anteriores.

```
  10 REM REVERSI
  20 BORDER 3: PAPER 5: CLS
  30 POKE 23658,8: POKE 23609,255
  40 GO SUB 5000
  50 GO SUB 4000
1000 REM MOVIMIENTOS DE NEGRAS
1999 STOP
2000 REM MOVIMIENTOS DE BLANCAS
2999 STOP
4000 REM PANTALLA
4999 STOP
5000 REM GRAFICOS
5999 STOP
```

Como siempre, comenzaremos por los caracteres gráficos, pero esta vez habrá uno más. En el Reversi, la pieza saltada se cambia sin más por el color del oponente (no hay necesidad por lo tanto de cambiar el fondo), pero las piezas suelen ser más grandes, por ejemplo del tamaño de cuatro cuadros de carácter, no uno. Otra diferencia, consecuencia del mayor tamaño de las piezas, es que los cuadros del tablero pueden ser ahora de cuatro cuadrados de carácter enteros y no hay necesidad de cortar ninguno de ellos. Por lo tanto, ahora podemos pasar a diseñar piezas de cuatro cuadrados cada una.

Para hacerlo, dibujar a tinta en una hoja de papel cuatro conjuntos de 8 × 8 cuadrados adyacentes en una configuración cuadrada. Incluimos un pequeño programa, numerado a partir de 9000, que sirve para presentar en la pantalla un ejemplo de la configuración descrita.

```
9010 FOR X=0 TO 128 STEP 64: PLOT 32,X+24:
     DRAW 128,0: NEXT X 9020 FOR X=0 TO 128
     STEP 64: PLOT 32+X,24: DRAW 0,128: NEX
     T X
9030 FOR A=0 TO 128 STEP 3: FOR X=0X=0 TO 1
     28 STEP 8: PLOT 32+A,24+X: NEXT X: NEX
     T A
9040 FOR A=0 TO 128 STEP 3: FOR X=0 TO 128
     STEP 8: PLOT 32+X,24+A: NEXT X: NEXT A
9050 PRINT OVER 1;AT 0,4;1;AT 0,12;1;AT 1,4
     ;2631;AT 1,12;2631
9060 PRINT OVER 1;AT 2,4;"8426842184268421"
```

Se han añadido un par de líneas que ponen los valores de las columnas en la cabecera del cuádruple cuadrado. Una vez escritas esas líneas, introducir RUN 9000 y saldrá en la pantalla el ejemplo programado (figura 5.1). Las líneas de puntos tardan en formarse, pero el resultado merece la pena. Dibujar el diagrama a tinta de esta forma, dejando a los lados suficiente espacio para escribir números de tres cifras.

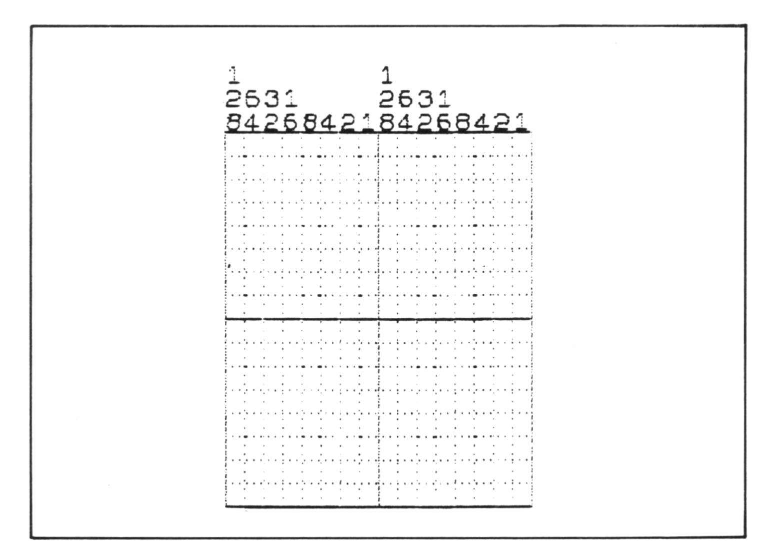


Figura 5.1

En esa configuración hay que dibujar un círculo inscrito en los cuatro cuadrados pero conservando el fondo alrededor. Como el carácter gráfico borra las líneas superior y lateral izquierda de cada gráfico, hay que añadir esa línea a cada carácter. El diseño de todo esto llevó mucho tiempo pero al final se dio con la solución, incluida en las líneas DATA del listado.

Esas líneas asocian los caracteres gráficos a las teclas A, B, C y D (para las fichas blancas). Para representar las fichas negras basta rellenar los círculos y calcular los nuevos números DATA que lo hagan. A tal efecto hemos añadido a la sección 9000 unas pocas líneas que muestran la forma de desarrollar los caracteres gráficos asociados a las letras A y B.

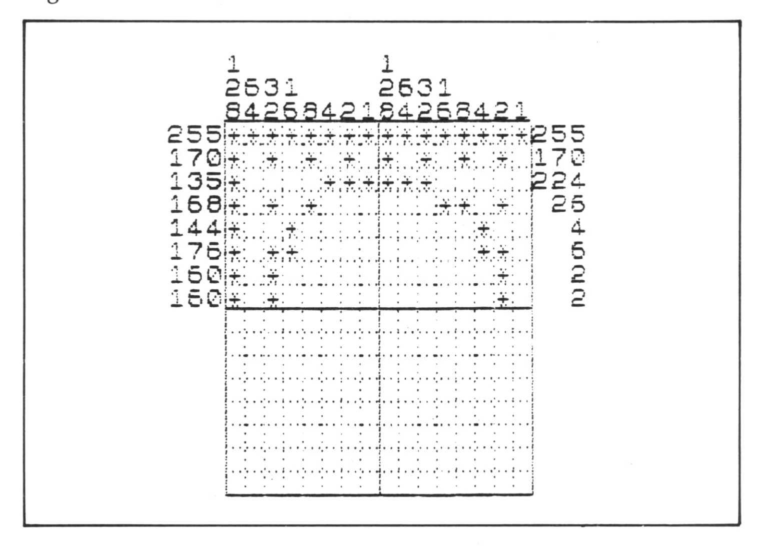
```
9070 PRINT OVER 1;AT 3,1;"255**************
     **255";AT 4,1;"170* * * * * * * * 170"
9080 PRINT OVER 1;AT 5,1;"135*    ******
        224";AT 6,1;"168* * *      ** *  26"
```

```
9090 PRINT OVER 1;AT 7,1;"144*  *           *
        4";AT 8,1;"176* **          **   6"
9100 PRINT OVER 1;AT 9,1;"160* *
     *   2";AT 10,1;"160* *          *   2"
```

Obsérvese la forma de empleo de OVER 1 en esas líneas. Se pone a continuación de PRINT y no es necesario repetirla en cada una de las instrucciones AT de una línea. Sólo sería preciso volver a ponerla cuando un signo de dos puntos ponga fin a la instrucción, y entonces habría que escribir también PRINT. Introducir RUN 9000 otra vez.

Este ejemplo (figura 5.2) muestra cómo se han desarrollado los caracteres A y B y cómo se han numerado las filas. Las fichas negras se hacen llenando los círculos blancos, borrando los números de fila e introduciendo los nuevos números de fila. El gráfico "I" que hace las funciones de tablero es muy sencillo: está constituido por filas alternas de 0 y 170 que definen una sucesión alterna de puntos y espacios. Con

Figura 5.2



esto quedan escritas todas las líneas de definición de los gráficos (sección 5000).

```
5010 FOR X=0 TO 8: FOR Y=0 TO 7: READ Z: PO
     KE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
5020 DATA 255,170,135,168,144,176,160,160,2
     55,170,224,26,4,6,2,2,160,160,160,176,
     144,172,131,170: REM AABBCC
5030 DATA 2,2,2,6,4,26,224,170,255,170,131,
     175,159,191,191,191,255,170,224,250,25
     2,254,254,254: REM DDEEFF
5040 DATA 191,191,191,191,159,175,131,170,2
     54,254,254,254,252,250,224,170,0,170,0
     ,170,0,170,0,170: REM GGHHII
5998 RETURN
```

Una vez introducido todo esto (activando el modo gráfico para escribir la segunda letra de cada instrucción REM), se puede ejecutar el programa para ver en las propias instrucciones REM el resultado de las entradas DATA. Se verá que la "I" es un cuadrado de líneas de puntos de un tono verde sobre fondo cian.

Este efecto fue mencionado ya en el capítulo 3, cuando se definió un cuadrado similar sobre un fondo amarillo. En aquel caso el cuadrado era el resultado de series alternas de 85, 0, 85, 0, etc. y esta vez es el resultado de series alternas de 0, 170, 0, 170, etc., comenzando ahora por el 0. Es justamente la alternancia del número de fila 0 y de un determinado número de fila lo que produce ese efecto; en tal sentido, la diferencia entre 85 y 170 como número de fila es precisamente ésta: 85 = 01010101 y 170 = 10101010.

Las aplicaciones de estos colores son considerables, por lo que dejaremos unos momentos el programa y desarrollaremos un ejemplo que asignaremos a los números de línea 8000 en adelante. Hacer una pequeña alteración en la línea 5010 (cam-

biar X=0 TO 8 por X=0 TO 12) y añadir DATA a la línea 5050. Este nuevo listado muestra las citadas alteraciones.

```
5010 FOR X=0 TO 12: FOR Y=0 TO 7: READ Z: P
     OKE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
5050 DATA 85,0,85,0,85,0,85,0,0,85,0,85,0,8
     5,0,85,170,0,170,0,17,0,0,170,0,0,170,
     0,170,0,170,0,170: REM JJKKLLMM
8000 REM PRUEBA DEL COLOR
8010 PAPER 6: CLS
8030 PRINT AT 5,3;"JJ KK LL MM";AT 6,3;"JJ
     KK LL MM"
8999 STOP
```

Los gráficos de la línea 5030 son: segunda J, segunda K, segunda L y segunda M. Ejecutar todo el programa y cuando al pie de la pantalla salga el aviso STOP, introducir RUN 8000. En la pantalla, ahora amarilla, saldrán cuatro cuadrados de puntos de colores alternos: verde, rojo, verde, rojo. Se verá que el empleo del 85 en vez del 170 tiene muy pocas consecuencias: sólo una ligera diferencia en la altura de la pantalla causada por la disposición de un 0 al principio o al final. Cambiar ahora el PAPER 6 de la línea 8010 por PAPER 5 (cian) e introducir otra vez RUN 8000. Esta vez los cuadrados son: azul, verde, azul, verde.

Por ahora se pueden dejar en el programa las alteraciones hechas porque el aviso STOP de la línea 5999 hará que no tengan ningún efecto en la ejecución del programa. Pero quien esté satisfecho porque ahora entiende a la perfección cómo se consigue ese efecto, que borre las líneas 5050, 8000, 8010, 8030 y 8999 y que cambie el 12 de la línea 5010 por un 8.

Habiendo completado la sección dedicada a los gráficos, sección 5000, podemos pasar a componer la pantalla, sección 4000. Tenemos asociado a la letra "I" el carácter que hace las

funciones de fondo del tablero, luego ésa será la primera línea, seguida por la definición de las letras y líneas gruesas del tablero.

```
4010 FOR X=2 TO 17: FOR Y=2 TO 17: PRINT AT
     X,Y;"I": NEXT Y: NEXT X
4020 FOR X=0 TO 128 STEP 16: PLOT 16+X,31:
     DRAW 0,128: NEXT X
4030 FOR Y=0 TO 128 STEP 16: PLOT 16,31+Y:
     DRAW 128,0; NEXT Y
4040 PRINT AT 1,2;"A B C D E F G H";AT 2,1;
     "A";AT 4,1;"B";AT 6,1;"C";AT 8,1;"D";A
     T 10,1;"E";AT 12,1;"F";AT 14,1;"G";AT
     16,1;"H"
4050 PRINT INVERSE 1; INK 1;AT 19,1;"  R E
     V E R S I  "
```

La letra "I" de la línea 4010, introducida en el modo gráfico, define el cuadrado de puntos que constituye el tablero. Las letras A a la H (asignadas a los cuadros del tablero) no quedan bien centradas con respecto a ellos, pero eso es algo que no se puede evitar pues cada cuadro grande está definido por cuatro posiciones PRINT AT y no es posible imprimir una letra a caballo entre dos posiciones. Obsérvese que en la línea 4040 hay un solo espacio entre las letras A, B, C, etc., mientras que en la línea 4050 hay también un espacio entre cada letra de la palabra REVERSI pero antes y después de ella hay dos espacios. La ejecución del programa debe producir la presentación de la pantalla mostrada en la figura 5.3.

En caso contrario, repasar las entradas. Se ve a la izquierda de la pantalla un tablero grande y debajo el título del juego. Queda a la derecha de la pantalla un espacio suficientemente amplio para incluir las instrucciones del juego, que es lo que haremos a continuación. Además, la partida de Reversi se suele abrir con cuatro fichas (dos blancas y dos negras) en el centro del tablero. Las instrucciones que cubren estos dos puntos componen el listado siguiente.

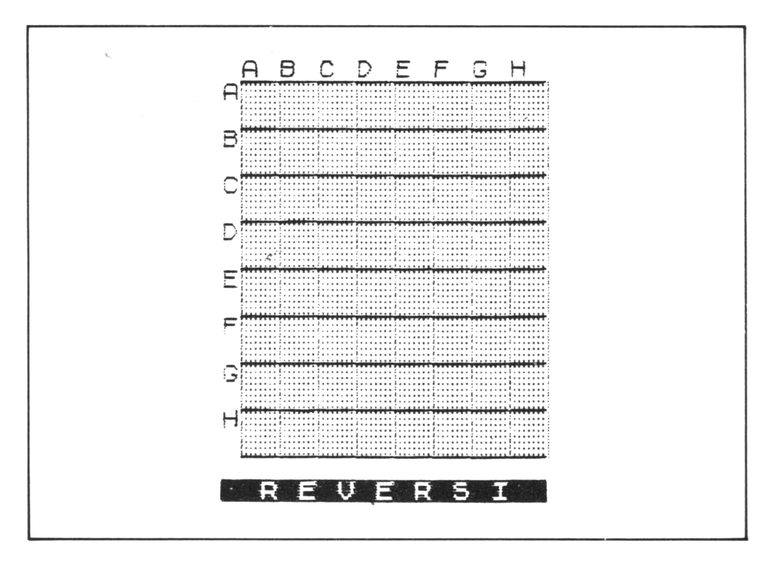


Figura 5.3

```
4060 PRINT AT 2,19;"SALTE SOBRE";AT 3,19;"E
     L OPONENTE";AT 4,19;"EN DIAGONAL,";AT
     5,19;"HORIZONTAL,"
4070 PRINT AT 6,19;"O VERTICAL.";AT 7,19;"-
     ------------";AT 8,19;"CAMBIE POR";AT
     9,19:"SU COLOR."
4080 PRINT AT 10,19;"-------------";AT 11,1
     9;"FIN DEL JUEGO";AT 12,19;"CUANDO NO
     SE";AT 13,19;"PUEDE SALTAR."
4090 PRINT AT 14,19;"--------------";AT 15,
     19;"GANA QUIEN";AT 16,19;"TIENE MAS";A
     T 17,19;"FICHAS."
4100 PRINT AT 8,8;"EF";AT 9,8;"GH";AT 10,8;
     "AB";AT 11,8;"CD"
4110 PRINT AT 10,10;"EF";AT 11,10;"GH";AT 8
     ,10;"AB";AT 9,10;"CD"
4150 RETURN
```

La línea 4100 programa los gráficos AB, CD, EF y GH, y la misma secuencia se repite en la línea 4110. Si se ejecuta el programa ahora se verá que la sección dedicada a la pantalla está completa. Pasemos por lo tanto a la sección dedicada a los movimientos de las negras, sección 1000. La primera línea debe convertir las letras A, B, C, D del borde superior y del lateral izquierdo de la pantalla en números X,Y, los números que el ordenador reconoce. Una línea LET A=2: LET B=4: etc. lo hará. Todo está ya a punto para una INPUT.

En el Reversi los movimientos son sencillos pues no hay eliminación de piezas sino sustitución de una por otra. Hay otro punto: tras algunos movimientos es posible cambiar una segunda pieza del oponente (o incluso más) en la misma tirada. Por lo tanto, debe haber una tercera INPUT a tal efecto. Programar todo esto es muy fácil, y sólo ocupa ocho líneas.

```
1010 LET A=2: LET B=4: LET C=6: LET D=8: LE
     T E=10: LET F=12: LET G=14: LET H=16
1020 INPUT "NEGRA A (LATERAL/SUPERIOR):";X;
     Y
1030 PRINT AT X,Y;"EF";AT X+1,Y;"GH"
1040 INPUT "CAMBIA BLANCA (LAT./SUP.):";U;V
1050 PRINT AT U,V;"EF";AT U+1,V;"GH"
1060 INPUT INVERSE 1; FLASH 1;"PUEDE CAMBIA
     R OTRA? (S/N):";Z$
1070 IF Z$="S" THEN GO TO 1040
1080 IF Z$="N" THEN GO TO 2000
```

Los gráficos de las líneas 1030 y 1050 son EF y GH en ambos casos. Obsérvese atentamente que la segunda parte de la instrucción PRINT AT de las líneas 1030 y 1050 ponen la segunda mitad de la ficha en la fila inmediatamente inferior a la de la primera mitad mediante X+1 y U+1.

Estos son los únicos movimientos de las negras. Ahora pasaremos a escribir los movimientos de las blancas EDITando la sección 1000 y haciendo las modificaciones necesarias, los

colores y las instrucciones GO TO, para convertirla en la sección 2000.

```
2010 LET A=2: LET B=4: LET C=6:LET D=8: LET
     E=10: LET F=12: LET G=14: LET H=16
2020 INPUT "BLANCA A (LATERAL/SUPERIOR):";X
     ;Y
2030 PRINT AT X,Y;"AB";AT X+1,Y;"CD"
2040 INPUT "CAMBIA NEGRA (LAT./SUP.):";U;V
2050 PRINT AT U,V;"AB";AT U+1,V;"CD"
2060 INPUT INVERSE 1; FLASH 1;"PUEDE CAMBIA
     R OTRA? (S/N):";Z$
2070 IF Z$="S" THEN GO TO 2040
2080 IF Z$="N" THEN GO TO 1000
```

Los gráficos de las líneas 2030 y 2050 son AB y CD en ambos casos. Con estas líneas el juego está completo y listo para ser jugado por dos jugadores. Al ejecutar el programa saldrá en la pantalla un tablero similar al mostrado en la figura 5.4.

Para los que no hayan jugado nunca al Reversi hay que decir que todos los movimientos se hacen saltando sobre una ficha del oponente en cualquier dirección (sólo sobre una) y luego cambiando esa ficha saltada por una propia. Una vez dado el salto y cambiada la ficha del oponente, no es extraño que se pueda cambiar otra más. Se puede cambiar cualquier ficha del oponente que esté escoltada a uno y otro lado por los colores propios (horizontalmente, verticalmente o diagonalmente), y puede darse el caso de que un solo movimiento dé derecho a cambiar muchas fichas del oponente. Veamos los siguientes ejemplos de situaciones a las que se ha llegado tras algunos movimientos, figura 5.5.

Las blancas han movido a F,D y han cambiado la negra de E,D. Las blancas pueden cambiar también la negra de E, E, y así sucesivamente. Después les toca a las negras.

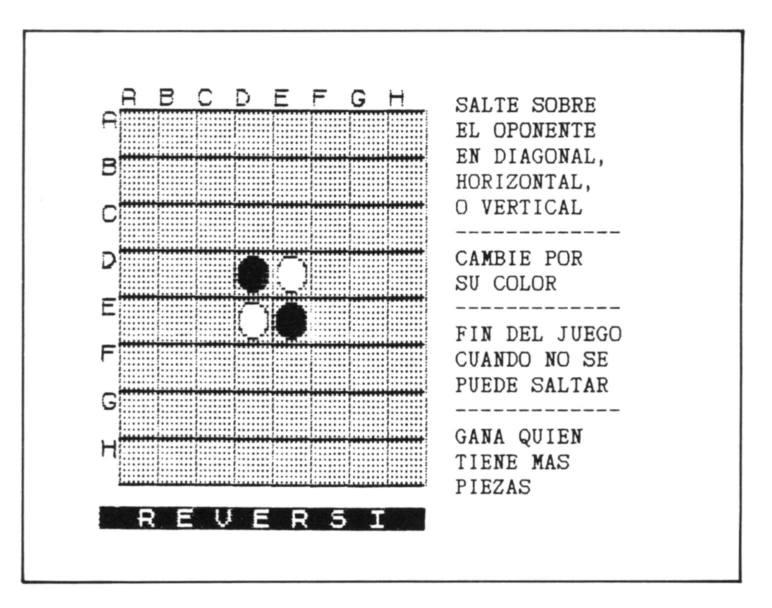
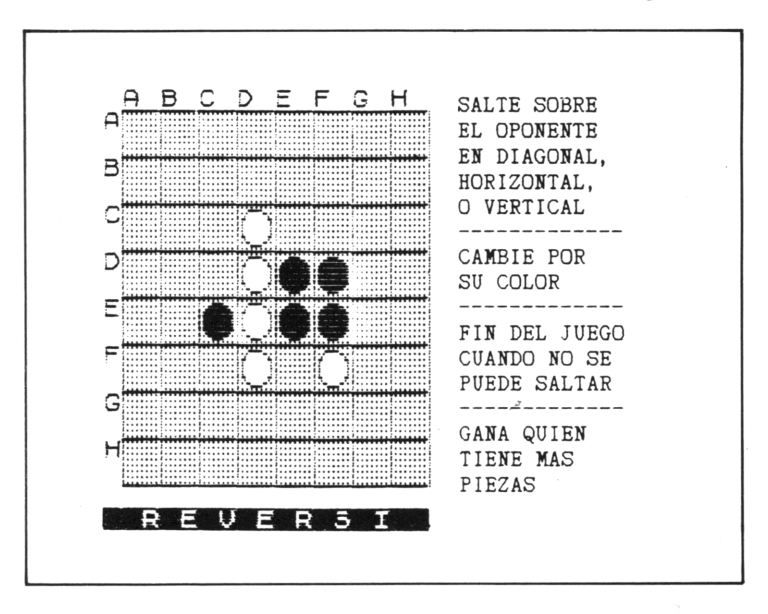


Figura 5.4 *Figura 5.5*



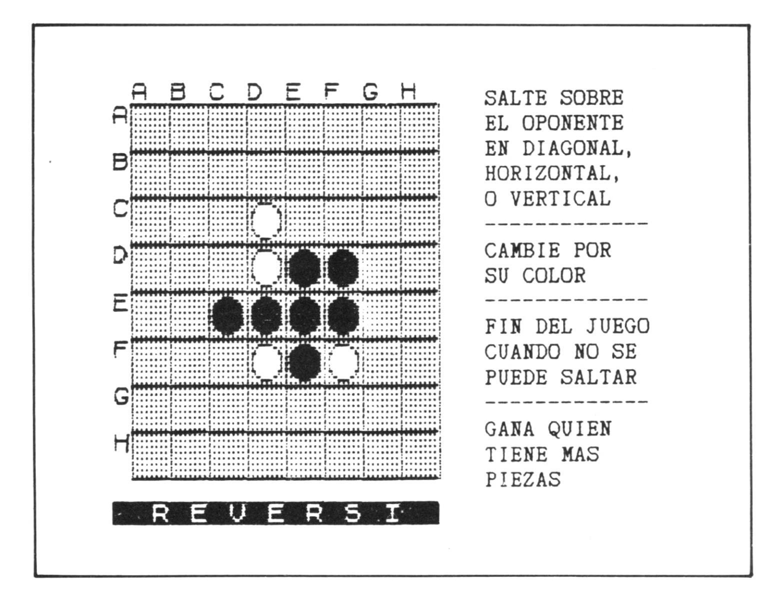
Las negras saltan a E,F abriendo la posibilidad de cambiar las blancas de E,E y E,D. La posición de las blancas pasa a ser la mostrada en la figura 5.6.

Las blancas están en una mala posición pues sólo pueden saltar a dos posiciones. Pero da la casualidad de que una vez elegida una (D,D a F,B) pueden cambiar las negras de E,C, E,D, E,E y F,E. La posición ha pasado a ser la de la figura 5.7.

Estos pocos movimientos son suficientes sin embargo para ver por qué se llama a este juego REVERSI. Queda por mostrar una situación en que no se puede dar ningún salto, figura 5.8.

Esta figura muestra que las negras no tienen ninguna posibilidad de hacer un movimiento. Sólo la ficha de D,F está escoltada por una blanca pero en todas las direcciones hay más

Figura 5.6



A B C D E F G H

SALTE SOBRE
EL OPONENTE
EN DIAGONAL,
HORIZONTAL,
O VERTICAL

CAMBIE POR
SU COLOR

FIN DEL JUEGO
CUANDO NO SE
PUEDE SALTAR

GANA QUIEN
TIENE MAS
PIEZAS

R E V E R S I

Figura 5.7

Figura 5.8

A B C D E F G H

SALTE SOBRE
EL OPONENTE
EN DIAGONAL,
HORIZONTAL,
O VERTICAL

CAMBIE POR
SU COLOR

FIN DEL JUEGO
CUANDO NO SE
PUEDE SALTAR

GANA QUIEN
TIENE MAS
PIEZAS

R E V E R S I

de una blanca en línea, luego no hay ninguna blanca única sobre la que pueda saltar una negra. La partida ha llegado a su fin con el triunfo de las blancas, simplemente porque son más.

El juego de Reversi que se acaba de escribir está ya completo. Borrar las líneas STOP 1999, 2999 y 4999. Si se van a dejar los dos ejemplos asignados a las líneas 8000 y 9000 en adelante, mantener la línea 5999 STOP pues en caso contrario serían ejecutadas también.

6. EL ANTIGUO JUEGO DEL GO

Debo confesar que siempre me he sentido fascinado por el juego del GO, y tanto es así que al poco tiempo de tener mi primer Spectrum escribí un programa para jugar al GO. Normalmente se juega con piezas negras y blancas (llamadas "piedras") que se colocan en las intersecciones de las líneas de una cuadrícula. El tablero tiene 19 líneas verticales y 19 horizontales, haciendo un total de 361 intersecciones (llamadas "puntos"). Los tableros son nòrmalmente de $15\frac{1}{4} \times 17\frac{1}{4}$ pulgadas (368×436 mm), bastante voluminosos por lo tanto. Las piedras (unas 180 de cada color aproximadamente) cambian mucho de posición durante el transcurso del juego. Por eso ideé una versión del juego a base de un tablero taladrado y muchos alfileres pequeños con cabezas de color negro y blanco en lugar de piedras.

Sin embargo, cuando escribí el programa del GO para el ZX Spectrum descubrí que es mucho más cómodo para jugar, y ahora lo tengo en la pantalla con mucha frecuencia, para jugar con algún amigo o para estudiar nuevas estrategias. En la presente versión el ordenador no toma parte; es para dos jugadores.

El programa es relativamente corto pues los movimientos consisten en poner o quitar piezas. Sólo hay que programar cuatro caracteres gráficos, una pantalla del juego, y los movimientos. Por lo tanto, el esqueleto o plan resulta bastante fácil.

```
  10 REM * * G O * *
  20 POKE 23658,8: POKE 23609,255
  30 BORDER 4: PAPER 7: CLS
  40 GOSUB 3000
  50 GO SUB 2000
1000 REM * * MOVIMIENTOS * *
1999 STOP
2000 REM * * PANTALLA * *
2999 STOP
3000 REM * * GRAFICOS * *
3999 STOP
```

Los cuatro caracteres gráficos que necesitamos son: una piedra blanca, una piedra negra, una cruz de recambio para las intersecciones (puntos) y una X para nueve puntos del tablero. Los conocimientos adquiridos en los anteriores capítulos deben haber capacitado al lector para diseñar estos gráficos sencillos, pero he aquí de todas formas los números DATA de los gráficos que hemos diseñado.

```
3010 FOR X=0 TO 3: FOR Y=0 TO 7: READ Z: PO
     KE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
3020 DATA 0,60,66,66,66,66,60,0,0,60,126,12
     6,126,126,60,0,8,8,8,255,8,8,8,8,42,
     28,255,28,42,8,8: REM AABBCCDD
3030 RETURN
```

Dibujar el tablero incorporando letras y números de posición es algo más complicado, pero hay una forma convencional de incluir la numeración. Las letras del borde inferior (A a la T) son bastante fáciles y se las puede asignar números de la forma usual (LET A=2:, LET B=3:, etc.), pero los nú-

meros se extienden por el lado izquierdo del tablero en orden ascendente, no descendente como los números X, luego es preciso incluir una fórmula especial que convierta los números correspondientes a las líneas en los números X que utiliza el ordenador.

Afortunadamente, los números ascienden a razón de uno por cada número X, luego si se comienza asignando el número 19 a la posición X identificada por el ordenador como 1, podemos utilizar la instrucción LET X=20 – X, que haría corresponder a ese 19 el número X 1. Más adelante se aclarará este punto.

Primero hay que dibujar las líneas del tablero e incluir una X en nueve puntos. Estas X no son importantes para el desarrollo normal del juego pero dan una idea de la posición. Por ejemplo, si se decide jugar un cuarto de partida (algo normal en partidas cortas), la X central constituirá una de las esquinas del nuevo tablero (de tamaño un cuarto del normal), delimitado por "1 a 10" y "A a K" o "1 a 19" y "A a K", etc. Cuando durante el transcurso del juego algunas de esas X quedan borradas por piedras, no son reemplazadas aunque después se quiten las piedras de esos sitios.

```
2010 FOR X=0 TO 144 STEP 8: PLOT 17,X+20: D
     RAW 147,0: NEXT X
2020 FOR X=0 TO 144 STEP 8: PLOT X+20,18: D
     RAW 0,146: NEXT X
2030 PRINT INK 1;AT 1,0;19'18'17'16'15'14'1
     3'12'11'10: PRINT INK 1;AT 11,1;9;AT 1
     2,1;8;AT 13,1;7;AT 14,1;6;AT 15,1;5;AT
     16,1;4;AT 17,1;3;AT 18,1;2;AT 19,1;1
2040 PRINT INK 1;AT 20,2;"ABCDEFGHJKLMNOPQR
     ST"
2050 PRINT AT 4,5;"D";AT 4,11;"D":AT 4,17;"
     D";AT 10,5;"D";AT 10,11;"D";AT 10,17;"
     D";AT 16,5;"D";AT 16,11;"D";AT 16,17;"
     D"
```

Obsérvese que no se utiliza la letra I en la fila de letras de posición del borde inferior del tablero. Se hace así para evitar confusiones con el número 1. A la letra D queda asociado el carácter gráfico que hace las funciones de X y a la letra C queda asociada la cruz normal. Obsérvese también que las líneas que forman el tablero deben prolongarse ligeramente hasta la proximidad de números de la izquierda y las letras del pie del tablero. Así será más fácil leer las posiciones de los puntos durante el transcurso del juego. Las líneas definidas por 2010 y 2020 están organizadas de forma que se crucen en el centro de un espacio de carácter con objeto de que cuando se introduzcan e imprimen piedras durante el transcurso del juego, éstas caigan en la posición correcta.

Faltan ahora el título del juego y unas pocas instrucciones. Están en el siguiente listado, que completa la sección 2000.

```
2060 PRINT INVERSE 1; INK 1;AT 0,5;" * * G
     O * * "
2070 PRINT AT 0,21;"PUNTOS N:";AT 1,21;"PUN
     TOS B:"
2080 PRINT INK 1;AT 3,21;"NEGRAS SALEN"
2090 PRINT AT 4,21;"INTRODUCIR";AT 5,21;"NO
       LAT.";AT 6,21;"Y DESPUES";AT 7,21;"L
     ETRA INF."
2100 PRINT INK 1;AT 8,21;"SEPARADAS"
2110 PRINT INK 1;AT 12,21;"TODAS LAS";AT 14
     ,22;"ENTRADAS";AT 16,23;"SEPARADAS"
2500 RETURN
```

Ejecutar el programa ahora. Saldrá en la pantalla una imagen similar a la de la figura 6.1. Las instrucciones incluidas en el lado derecho sirven para recordar a los jugadores que todas las entradas son dobles: NUMERO-ENTER, LETRA-ENTER.

Ahora vienen los "movimientos", por lo que pasamos a la línea 1000, que rige las entradas de las negras. Las entradas

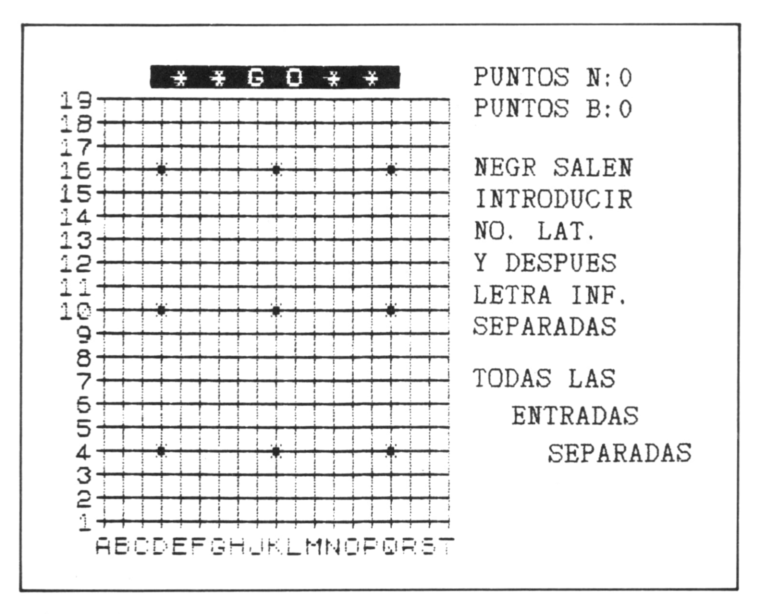


Figura 6.1

de las blancas están en la 1200; las eliminaciones de las blancas en la 1500, y las eliminaciones de las negras en la 1600. La sección constituida por la línea 1500 y siguientes contendrá los marcadores, sobre los que hablaremos largo y tendido más adelante. La variable VN es el marcador de las negras y la VB el de las blancas.

Hay que incluir algunas instrucciones que asignen números Y a las letras, y pensar alguna solución para el problema de los números del lateral izquierdo. Podemos utilizar LET X=20 – X, pero X no cuenta como una verdadera variable porque aquí forma parte de una función matemática. Por lo tanto, será preciso poner esta instrucción antes de cualquier instrucción que contenga PRINT AT X,Y con objeto de restablecer la función, lo cual significa que hay que ponerla cuatro veces. Haciéndolo así no hay ningún problema.

```
1010 LET VN=0: LET VB=0: PRINT AT 0,29;VN;A
     T 1,29;VB
1020 LET X=20-X
1030 LET A=2: LET B=3: LET C=4: LET D=5: LE
     T E=6: LET F=7: LET G=8: LET H=9: LET
     J=10
1040 LET K=11: LET L=12: LET M=13: LET N=14
     : LET O=15: LET P=16: LET Q=17: LET R=
     18: LET S=19: LET T=20
1050 INPUT "NEGRA A (NO./LET.)";X;Y
1060 LET X=20-X
1070 PRINT AT X,Y;"B"
1080 INPUT FLASH 1;"ELIMINA BLANCA? (S/N):"
     ;B$
1090 IF B$="S" THEN GO TO 1500
1100 IF B$="N" THEN GO TO 1200
```

Los movimientos de las negras están regidos por las líneas 1050 a 1100. Se habrá advertido la presencia de un signo intermitente en la pregunta INPUT “ELIMINA BLANCA? (S/N)”. Se ha hecho así porque en la excitación del juego es fácil pasar por alto esa pregunta y dar paso al movimiento del otro color antes de que la INPUT correspondiente lo diga. Los movimientos de las blancas regidos por las líneas 1200 en adelante son una simple repetición de los movimientos de las negras, con las necesarias alteraciones del color y de los números GO TO.

```
1200 INPUT "BLANCA A (NO./LET.)";X;Y
1210 LET X=20-X
1220 PRINT AT X,Y;"A"
1230 INPUT FLASH 1;"ELIMINA NEGRA? (S/N):";
     B$
1240 IF B$="S" THEN GO TO 1600
1250 IF B$="N" THEN GO TO 1050
```

Pasemos ahora a las eliminaciones de piedras. Son muy fáciles, con la única complicación de que incluyen también un

cambio del marcador de la esquina superior derecha de la pantalla. El siguiente listado cubre esos puntos.

```
1500 REM ELIMINACIONES
1510 INPUT "ELIMINA BLANCA (NO./LETRA):";X;
     Y
1520 LET X=20-X
1530 PRINT AT X,Y;"C": LET VN=VN+1: PRINT A
     T 0,29;VN
1540 INPUT FLASH 1;"MAS? (S/N):";C$
1550 IF C$="S" THEN GO TO 1510
1560 IF C$="N" THEN GO TO 1200
1610 INPUT "ELIMINA NEGRA (NO./LETRA):";X;Y
1620 LET X=20-X
1630 PRINT AT X,Y;"C": LET VB=VB+1: PRINT A
     T 0,29;VB
1640 INPUT FLASH 1;"MAS? (S/N):";C$
1650 IF C$="S" THEN GO TO 1610
1660 IF C$="N" THEN GO TO 1050
```

Si se ejecuta el programa ahora se verá que funciona bien, pero las INPUT que hacen preguntas necesitan hacerse notar más. Un pitido (BEEP) podría constituir una buena solución. La línea a incluir a estos efectos es: 1075 FOR X=10 TO 20: BEEP .1,X: NEXT X. Con esta línea, el programa da una muy buena impresión, por lo que si se quiere se puede incluir también en las líneas 1225, 1535 y 1635. Esta adición añade una precaución más al obligar al jugador a prestar atención a las preguntas INPUT entre movimiento y movimiento.

```
1075 FOR X=10 TO 20: BEEP .1,X: NEXT X
1225 FOR X=10 TO 20: BEEP .1,X: NEXT X
1535 FOR X=10 TO 20: BEEP .1,X: NEXT X
1635 FOR X=10 TO 20: BEEP .1,X: NEXT X
```

Estas adiciones ponen bien claramente de manifiesto la conveniencia de numerar las líneas de los programas de 10 en

10 porque durante la escritura de un programa brota con no poca frecuencia la necesidad de añadir una o más líneas a lo ya escrito y esa separación permite hacerlo.

Ya hemos mencionado anteriormente los marcadores. Los presentados en la pantalla son sólo una parte de los puntos, y además negativa. Al final de la partida (por acuerdo mutuo o porque un jugador no encuentra ningún lugar seguro para una piedra), se cuentan todos los "puntos" rodeados por piedras negras y esa cantidad son los puntos de las negras pero no sin antes restar la cantidad que muestra el marcador de las piedras negras capturadas por las blancas. Similarmente, la cantidad de puntos rodeados por piedras blancas constituye su marcador una vez restadas las piedras cogidas por las negras. En resumen: puntos final = puntos rodeados menos piedras capturadas.

El GO es un juego muy complejo y las reglas son demasiado extensas para darlas aquí. Lo mejor es hacerse con alguno de los muchos libros de GO publicados.

Para dar a los lectores la oportunidad de jugar un poco mientras esperan la llegada de su libro de las reglas del juego, incluimos aquí unas cuantas indicaciones. El objetivo es ocupar, rodeando, más territorio (puntos) que el oponente. Las negras salen poniendo una piedra en cualquier sitio y luego juegan las blancas. Todas las piedras permanecen en los mismos sitios a no ser que sean rodeadas. Estar rodeado significa tener piedras del oponente en todos los puntos que rodean a una o más piedras propias, en la misma línea, no diagonalmente. Lo dicho necesita aclaración y por eso hemos escrito un pequeño programa con ejemplos, numerado a partir de la línea 5000. Se verá que utiliza algunas de las líneas escritas anteriormente (2010 en adelante).

```
5000 REM EJEMPLO
5010 FOR X=0 TO 144 STEP 8: PLOT 17,X+20: D
     RAW 147,0: NEXT X
```

```
5020 FOR X=0 TO 144 STEP 8; PLOTX+20,18: DR
     AW 0,146: NEXT X
5030 PRINT INK 1;AT 1,0;19'18'17'16'15'14'1
     3'12'11'10: PRINT INK 1;AT 11,1;9;AT 1
     2,1;8;AT 13,1;7;AT 14,1;6;AT 15,1;5;AT
     16,1;4;AT 17,1;3;AT 18,1;2;AT 19,1;1
5040 PRINT INK 1;AT 20,2;"ABCDEFGHJKLMNOPQR
     ST"
5050 LET A=2: LET B=3: LET C=4: LET D=5: LE
     T E=6: LET F=7: LET G=8: LET H=9: LET
     J=10
5060 LET K=11: LET L=12: LET M=13: LET N=14
     : LET O=15: LET P=16: LET Q=17: LET R=
     18: LET S=19: LET T=20
5070 PRINT AT 1,B;"A";AT 1,F;"AAA";AT 1,M;"
     AAA"
5080 PRINT AT 2,A;"ABA";AT 2,E;"ABBBA";AT 2
     ,L;"ABBBA"
5090 PRINT AT 3,B;"A";AT 3,E;"ABDBA";AT 3,L
     ;"ABDBA"
5100 PRINT AT 4,C;"A";AT 4,E;"ABBBA";AT 4,L
     ;"ABDBA"
5110 PRINT AT 5,F;"AAA";AT 5,L;"ABBBA"
5120 PRINT AT 6,A;"A";AT 6,M;"AAA"
5130 PRINT AT 7,A;"BA";AT 7,G;"D";AT 8,A;"A
     ";AT 8,N;"F";AT 9,E;"AAAAA"
5140 PRINT AT 10,A;"B";AT 10,D;"ABBBBBA"
5150 PRINT AT 11,D;"ABDBDBA";AT 11,N;"ADA"
5160 PRINT AT 12,D;"ABBBBBA";AT 12,N;"DBD"
5170 PRINT AT 13,E;"AAAAA";AT 13,N;"ADA"
5180 PRINT AT 15,G;"E";AT 15,O;"C"
```

RUN 5000 presenta en la pantalla los ejemplos A a F (figura 6.2). Este listado no interferirá con el programa principal.

A y B son ejemplos de una piedra negra perfectamente rodeada. Veamos el ejemplo C: esa piedra negra no está ro-

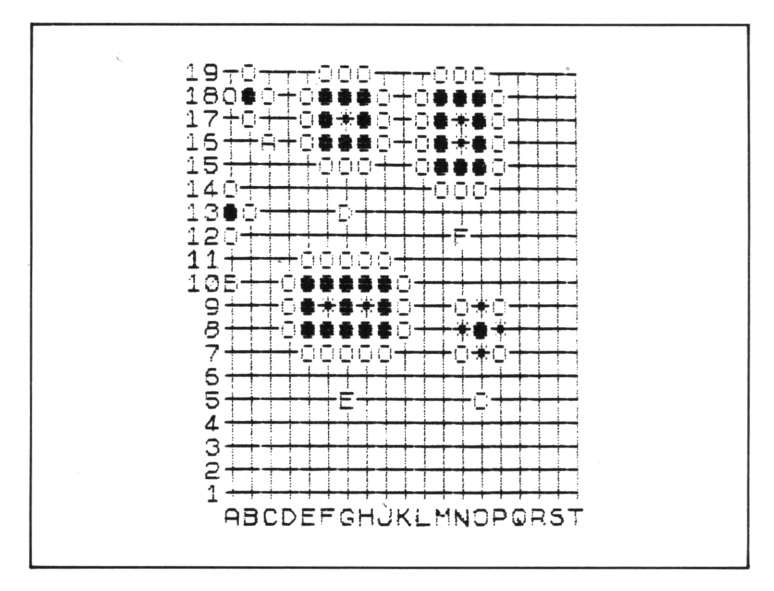


Figura 6.2

deada; las cruces muestran dónde debería haber piedras blancas para tener rodeada a la negra.

El ejemplo D muestra una parcela de piedras negras rodeadas correctamente por blancas. Las blancas están en disposición de poner una de sus piedras en el "ojo" central (marcado con una cruz) y hacerse con todas las negras, pues quedarían totalmente rodeadas por dentro y por fuera. Por lo tanto, dejar un solo "ojo". (que así es como se llaman esos puntos rodeados de piedras) no es bueno; la parcela así fortificada es muy vulnerable.

El ejemplo F muestra una situación muy similar al D, pero ahora hay dos "ojos" adyacentes. Esta situación es también vulnerable pues las blancas están en disposición de arrojar del campo de batalla a una negra colocando una piedra blanca en una de las posiciones marcadas con una cruz. Las negras pueden contrarrestar esta situación poniendo una piedra en la

otra cruz y eliminando la blanca recién puesta. Sin embargo, ahora la parcela negra tendría sólo un "ojo" y las blancas, una vez hecho otro movimiento porque así lo exigen las reglas, podrían poner una piedra en ese sitio y eliminar todas las negras.

El ejemplo E muestra la forma de dejar dos o más "ojos" individuales en una parcela. Estas parcelas son totalmente invulnerables y constituyen el principal objetivo del buen jugador.

Con estas pocas instrucciones se puede jugar al GO ya, pero hay muchas sutilezas que sólo se pueden sacar a la luz estudiando todas las reglas y analizando muchos ejemplos. Quien esté verdaderamente interesado en este juego, que compre un buen libro. Los ardientes seguidores del GO proclaman que es un juego de más profundidad que el ajedrez.

7. EL AJEDREZ, UN JUEGO DIFICIL

Escribir un programa para jugar al ajedrez acarrea problemas bastante complejos, incluso si, como en este caso, es sólo para dos jugadores (sin que participe el ordenador), ya que las piezas son diferentes y están dispuestas unas en cuadros blancos y otras en cuadros negros. En esta versión haremos los cuadros negros de color rojo y los blancos de color amarillo. Los movimientos son bastante complejos y, además, hay que diseñar por lo menos doce caracteres gráficos que representen a las distintas piezas: seis negras y seis blancas.

La primera tarea consiste por lo tanto en dividir el programa en secciones: los gráficos constituirán una sección; la composición de la pantalla otra; otra los movimientos de las blancas y otra más los movimientos de las negras; habrá una sección más encargada de reimprimir el fondo rojo tras cada movimiento; otra sección se encargará de mostrar las piezas comidas y otra más se encargará de la promoción de los peones a reinas cuando se dé el caso. Hay por lo tanto seis secciones como mínimo. Conviene mantener las diferentes secciones bien delimitadas con objeto de que sea fácil recordarlas durante la composición del programa; las pondremos a inter-

valos de 1000. Además, todo es mucho más sencillo cuando se puede desarrollar independientemente cada sección. En consecuencia, asignando la primera sección a los movimientos de las blancas, a partir de la línea 1000, resulta el siguiente listado del esqueleto o plan.

```
  10 REM ** AJEDREZ **
  20 POKE 23609,255: POKE 23658,8
  30 BORDER 2: PAPER 6: CLS
  40 GOSUB 7000
  50 GOSUB 6000
1000 REM MOVIMIENTOS DE BLANCAS
1999 STOP
2000 REM MOVIMIENTOS DE NEGRAS
2999 STOP
3000 REM REPOSICION DEL FONDO
3999 STOP
4000 REM IMPRESION DE LAS PIEZAS COMIDAS
4999 STOP
5000 REM PROMOCION DE PEONES
5999 STOP
6000 REM DISEÑO DE LA PANTALLA
6998 RETURN
6999 STOP
7000 REM CARACTERES GRAFICOS
7998 RETURN
7999 STOP
```

Una vez establecido el plan general podemos pasar a diseñar los caracteres gráficos. Las piezas de ajedrez que hemos hecho ocupan un espacio de un carácter en el caso del rey y un espacio relativamente más pequeño en el caso de las demás piezas. Hubiera sido mejor hacer piezas mucho más grandes, pero necesitamos ya doce caracteres gráficos para representar las seis piezas de cada jugador y el aumento de su tamaño hubiera supuesto diseñar más caracteres gráficos de los que tenemos permitidos por el momento. Hay además otro carácter gráfico encargado de la reposición del fondo, lo que hace

trece en total, por lo que la sección de generación de caracteres gráficos es ya suficientemente larga.

```
7010 FOR X=0 TO 12: FOR Y=0 TO 7: READ Z: P
     OKE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
7020 DATA 0,0,24,60,60,24,60,126,0,85,127,6
     2,62,62,127,127,0,24,52,110,126,60,126
     ,126: REM AA,BB,CC
7030 DATA 0,12,62,126,14,28,62,127,24,36,16
     3,90,189,126,126,126,24,126,219,219,90
     ,126,126,126: REM DD,EE,FF
7040 DATA 0,170,0,170,0,170,0,170: REM GG
7050 DATA 0,0,24,36,36,24,36,126,0,85,127,3
     4,34,34,99,127,0,24,44,82,66,36,102,12
     6, REM HH,II,JJ
7060 DATA 0,12,50,114,10,20,34,127,24,36,17
     1,90,189,102,90,126,24,126,153,153,90,
     126,90,126: REM KK,LL,MM
```

Si se ejecuta ahora el programa y se introduce después LIST, el listado mostrará los caracteres gráficos junto a las letras correspondientes de las instrucciones REM. Comprobar que todas las piezas están bien diseñadas y en caso contrario repasar las líneas DATA. Cuando todo sea correcto se puede pasar a la sección 6000.

Para representar en la pantalla un tablero de un tamaño razonable, cada cuadro debe ocupar el equivalente a cuatro espacios de carácter, pero, además, cada cuadro debe tener definido un espacio de un solo carácter situado en el centro en aptitud de recibir a una pieza. A tal efecto hay que dibujar líneas divisorias que partan por la mitad a los espacios de carácter, y por lo tanto los cuadros rojos deben estar definidos por líneas de puntos, no por un carácter gráfico. Se programaron las ocho filas correspondientes al carácter gráfico "G" después de dibujar la pantalla, y hubo que hacer muchas prue-

bas hasta conseguir que se correspondiera perfectamente con las líneas de puntos de la pantalla.

```
6010 FOR X=0 TO 128 STEP 16: PLOT X+20,28:
     DRAW 0,128: NEXT X
6020 FOR X=0 TO 128 STEP 16: PLOT 20,X+28:
     DRAW 128,0: NEXT X
6030 PRINT AT 19,3;"A B C D E F G H";AT 3,1
     ;"A";AT 5,1;"B";AT 7,1;"C";AT 9,1;"D";
     AT 11,1;"E";AT 13,1;"F";AT 15,1;"G";AT
     17,1;"H"
6040 FOR A=2 TO 14 STEP 2: FOR X=2 TO 14 ST
     EP 2: PLOT 20+A,28+X: PLOT 52+A,28+X:
     PLOT 84+A,28+X: PLOT 116+A,28+X: NEXT
     X: NEXT A
6050 FOR A=2 TO 14 STEP 2: FOR X=18 TO 30 S
     TEP 2: PLOT 36+A,28+X: PLOT 68+A,28+X:
     PLOT 100+A,28+X: PLOT 116+A,28+X: NEXT
     X: NEXT A
6060 FOR A=2 TO 14 STEP 2: FOR X=34 TO 46 S
     TEP 2: PLOT 20+A,28+X: PLOT 52+A,28+X:
     PLOT 84+A,28+X: PLOT 116+A,28+X: NEXT
     X: NEXT A
6070 FOR A=2 TO 14 STEP 2: FOR X=50 TO 62 S
     TEP 2: PLOT 36+A,28+X: PLOT 68+A,28+X:
     PLOT 100+A,28+X: PLOT 132+A,28+X: NEXT
     X: NEXT A
6080 FOR A=2 TO 14 STEP 2: FOR X=66 TO 78 S
     TEP 2: PLOT 20+A,28+X: PLOT 52+A,28+X:
     PLOT 84+A,28+X: PLOT 116+A,28+X: NEXT
     X: NEXT A
6090 FOR A=2 TO 14 STEP 2: FOR X=82 TO 94 S
     TEP 2: PLOT 36+A,28+X: PLOT 68+A,28+X:
     PLOT 100+A,28+X: PLOT 132+A,28+X: NEXT
     X: NEXT A
6100 FOR A=2 TO 14 STEP 2: FOR X=98 TO 110
     STEP 2: PLOT 20+A,28+X: PLOT 52+A,28+X
```

```
        : PLOT 84+A,28+X: PLOT 116+A,28+X: NEX
        T X: NEXT A
6110 FOR A=2 TO 14 STEP 2: FOR X=114 TO 126
        STEP 2: PLOT 36+A,28+X: PLOT 68+A,28+X
        : PLOT 100+A,28+X: PLOT 132+A,28+X: NE
        XT X: NEXT A
```

La ejecución de este listado producirá la impresión en pantalla de un tablero de ajedrez con cuadros rojos y amarillos, completado con letras impresas en los bordes inferior y lateral izquierdo. Se utilizan letras en vez de números, la solución más convencional, por la dificultad de convertir los números 1 a 8 (orden ascendente) en los números 3 a 17 STEP 2 (orden descendente), que son los números X. Por ejemplo, el 1 correspondería al número X 17, el 2 al número X 15 y el 3 al número X 13. La fórmula encargada de hacer esa conversión sería tremendamente complicada.

Las líneas de puntos que definen los cuadros rojos (series de puntos trazados con órdenes PLOT) tardan un cierto tiempo en extenderse por la pantalla pero sólo tienen que ser impresas una vez, al imprimir la pantalla, y durante el juego ya no cambian. Esa es la razón de que estén al final del programa, junto con la sección de generación de caracteres gráficos. El siguiente listado presenta en pantalla el título y algunas instrucciones del juego, así como la disposición inicial de las piezas y un lugar para imprimir las piezas comidas.

```
6120 PRINT INK 1;AT 1,20;"*********"; INVER
        SE 1;AT 2,20;" AJEDREZ ";INVERSE 0;AT
        3,20;"*********"
6130 PRINT INK 0;AT 5,20;"INTRODUCIR";AT 6,
        20;"LETR. LAT.";AT;7,20;"Y LET. INF.";
        AT 8,20;"SEPARADAS.";AT 9,20;"********
        *"
6140 PRINT INK 1;AT 10,20;"CODIGO:"
6150 PRINT AT 11,20;"PEON------P";AT 12,20;
        "TORRE-----T":AT 13,20:"CABALLO---C";A
```

```
T 14,20;"ALFIL-----A";AT 15,20;"REINA-
----Q";AT 16,20;"REY-------R"
6160 PRINT AT 17,3;"I";AT 17,5;"K";AT 17,7;
"J";AT 17,9;"L";AT 17,11;"M";AT 17,13;
"J";AT 17,15;"K";AT 17,17;"I"
6170 PRINT AT 15,3;"H";AT 15,5;"H";AT 15,7;
"H";AT 15,9;"H";AT 15,11;"H";AT 15,13;
"H";AT 15,15;"H";AT 15,17;"H"
6180 PRINT AT 5,3;"A";AT 5,5;"A";AT 5,7;"A"
;AT 5,9;"A";AT 5,11;"A";AT 5,13;"A";AT
5,15;"K";AT 5,17;"A"
6190 PRINT AT 3,3;"B";AT 3,5;"D";AT 3,7;"C"
;AT 3,9;"E";AT 3,11;"F";AT 3,13;"C";AT
3,15;"D";AT 3,17;"B"
6200 LET BL=0: LET NE=0
6210 PRINT AT 1,1;"BLANCAS COMIDAS:"BL;AT 2
0,1;"NEGRAS COMIDAS:";NE
```

Los caracteres gráficos (piezas) asociados a esta lista de letras se pueden ver en las instrucciones REM de las líneas DATA 7020 a 7060. Los *códigos* (P, T, C, A, Q y R) serán necesarios a la hora de jugar; se explican más adelante. Se puede ver en el listado anterior que las piezas blancas "comidas" han sido asignadas a la variable BL y, como han sido comidas por las negras, saldrán impresas detrás de la posición de las negras. Las negras comidas tienen asignada la variable NE, y saldrán impresas detrás de la posición de las blancas. Si se ejecuta ahora el programa, saldrá la pantalla completa (figura 7.1).

Esto completa las secciones 7000 y 6000. Ha llegado por lo tanto el momento de comenzar a escribir el juego propiamente dicho. Los movimientos precisarán una rutina del tipo DE/A, a continuación de la cual irá una INPUT que pida el *código*. El código es una letra, asignada a cada tipo de pieza, que hay que introducir para que el ordenador sepa qué pieza tiene que colocar en la posición de destino. Estos movimientos tendrán que incluir una subrutina que haga saber al ordenador

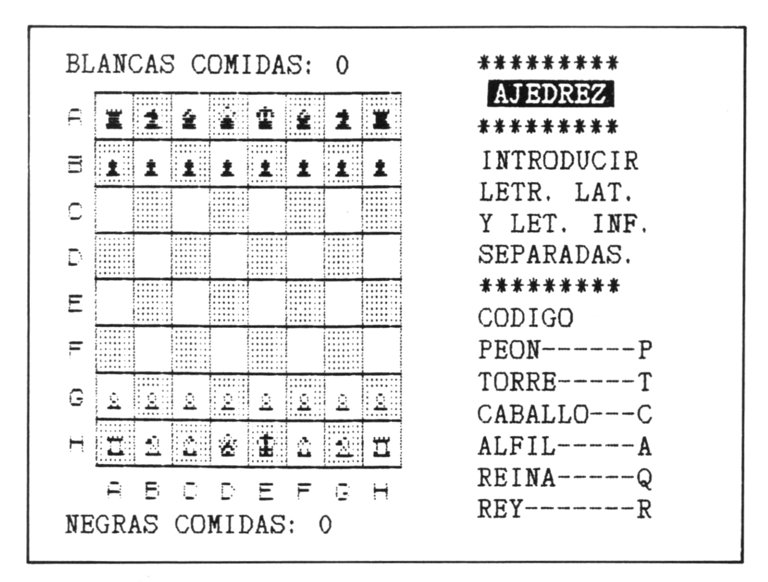


Figura 7.1

si tras quitar la pieza tiene que imprimir un espacio en blanco (en los cuadros amarillos) o el carácter gráfico "G" (en los cuadros rojos). La acción de esta subrutina tiene lugar al principio de la secuencia de movimientos, luego es mejor introducirla ahora. No olvidar que afecta a cuadros alternos en diferentes líneas, y por eso es bastante larga. Se introduce en la sección 3000.

```
3010 IF X+Y=A+B OR X+Y=A+D OR X+Y=A+F OR X+
     Y=A+H THEN GO TO 1050
3020 IF X+Y=B+A OR X+Y=B+C OR X+Y=B+E OR X+
     Y=B+G THEN GO TO 1050
3030 IF X+Y=C+B OR X+Y=C+D OR X+Y=C+F OR X+
     Y=C+H THEN GO TO 1050
3040 IF X+Y=D+A OR X+Y=D+C OR X+Y=D+E OR X+
     Y=D+G THEN GO TO 1050
3050 IF X+Y=E+B OR X+Y=E+D OR X+Y=E+F OR X+
     Y=E+H THEN GO TO 1050
```

```
3060 IF X+Y=F+A OR X+Y=F+C OR X+Y=F+E OR X+
     Y=F+G THEN GO TO 1050
3070 IF X+Y=G+B OR X+Y=G+D OR X+Y=G+F OR X+
     Y=G+H THEN GO TO 1050
3080 IF X+Y=H+A OR X+Y=H+C OR X+Y=H+E OR X+
     Y=H+G THEN GO TO 1050
3090 RETURN
```

Esta subrutina mira a ver si la posición X,Y (de la que se ha quitado la pieza) es roja o no. Si lo es, el ordenador va a la línea 1050, y si no lo es vuelve a la línea 1040, que es la línea siguiente a la GO SUB que le envió a la línea 3000 (en el siguiente listado). La misma subrutina servirá para las negras, pero ahora tendrá que enviar el ordenador a la sección de los movimientos de las negras, cuyo número aún no conocemos; lo dejaremos por lo tanto para más adelante. Los movimientos de las blancas están programados en este listado.

```
1005 LET A=3: LET B=5; LET C=7: LET D=9: LE
     T E=11: LET F=13: LET G=15: LET H=17
1010 INPUT "BLANCA DE:";X;Y
1020 INPUT "BLANCA A:";U;V
1030 GO SUB 3000
1040 PRINT AT X,Y;"G"
1060 INPUT "PIEZA MOVIDA? (P/T/C/A/Q/R):";J
     $
1070 IF J$="P" THEN PRINT AT U,V;"H"
1080 IF J$="T" THEN PRINT AT U,V;"I"
1090 IF J$="C" THEN PRINT AT U,V;"K"
1100 IF J$="A" THEN PRINT AT U,V;"J"
1110 IF J$="Q" THEN PRINT AT U,V;"L"
1120 IF J$="R" THEN PRINT AT U,V;"M"
1130 INPUT "PIEZA COMIDA? (S/N):";K$
1140 IF K$="N" THEN GO TO 1230
1150 IF K$="S" THEN INPUT "INTRODUCIR CODIG
     O (P/T/C/A/Q/R):";L$
1160 LET NE=NE+1
1170 IF L$="P" THEN PRINT AT 20,7+NE;"A"
```

```
1180 IF L$="T" THEN PRINT AT 20,7+NE;"B"
1190 IF L$="C" THEN PRINT AT 20,7+NE;"D"
1200 IF L$="A" THEN PRINT AT 20,7+NE;"C"
1210 IF L$="Q" THEN PRINT AT 20,7+NE;"E"
1220 IF L$="R" THEN PRINT AT 20,7+NE;"F"
1230 INPUT INVERSE 1; FLASH 1;"PROMOCION DE
     PEON? (S/N):";Q$
1240 IF Q$="N" THEN GO TO 2000
1250 IF Q$="S" THEN GO TO 5000
1260 GO TO 2000
```

Mientras escribía esta sección recordé que los peones son promovidos cuando llegan a la línea final de la base del oponente pero no sólo a reina sino que también pueden ser promovidos a torre, caballo o alfil, si así se desea. Parece conveniente prever esta posibilidad al programar este punto en la sección de movimientos. Ello implica la necesidad de responder a una pregunta INPUT más en cada movimiento, pero merece la pena. Además, parece conveniente utilizar esta sección de movimientos para incluir la exposición de las piezas comidas, con lo cual queda sin efecto la sección 4000.

Cuando acabé esta sección de los movimientos de las blancas todo funcionaba bien, y era por lo tanto el momento de introducir la segunda parte de la sección 3000 (movimientos de las negras), comenzando por la línea 3500, pues los movimientos de las negras iban a ser iguales que los movimientos de las blancas pero ocuparían la sección 2000 en vez de la 1000. En consecuencia, era posible poner GO TO 2050 en la sección 3500.

```
3510 IF X+Y=A+B OR X+Y=A+D OR X+Y=A+F OR X+
     Y=A+H THEN GO TO 2050
3520 IF X+Y=B+A OR X+Y=B+C OR X+Y=B+E OR X+
     Y=B+G THEN GO TO 2050
3530 IF X+Y=C+B OR X+Y=C+D OR X+Y=C+F OR X+
     Y=C+H THEN GO TO 2050
```

```
3540 IF X+Y=D+A OR X+Y=D+C OR X+Y=D+E OR X+
     Y=D+G THEN GO TO 2050
3550 IF X+Y=E+B OR X+Y=E+D OR X+Y=E+F OR X+
     Y=E+H THEN GO TO 2050
3560 IF X+Y=F+A OR X+Y=F+C OR X+Y=F+E OR X+
     Y=F+G THEN GO TO 2050
3570 IF X+Y=G+B OR X+Y=G+D OR X+Y=G+F OR X+
     Y=G+H THEN GO TO 2050
3580 IF X+Y=H+A OR X+Y=H+C OR X+Y=H+E OR X+
     Y=H+G THEN GO TO 2050
3590 RETURN
```

Ahora había que introducir todos los movimientos de las negras aprovechando los movimientos de las blancas como modelo e introduciendo las necesarias modificaciones. Se hizo EDITando todas las líneas 1050 en adelante y cambiando la primera cifra de los números de línea (el 1) por el 2, y haciendo luego las modificaciones necesarias en los textos de esas líneas, cuando los hubiera. Se hizo en un momento.

```
2005 LET A=3: LET B=5; LET C=7: LET D=9: LE
     T E=11: LET F=13: LET G=15: LET H=17
2010 INPUT "NEGRA DE:";X;Y
2020 INPUT "NEGRA A:";U;V
2030 GO SUB 3500
2040 PRINT AT X,Y;" ": GO TO 2060
2050 PRINT AT X,Y;"G"
2060 INPUT "PIEZA MOVIDA? (P/T/C/A/Q/R):";J
     $
2070 IF J$="P" THEN PRINT AT U,V;"A"
2080 IF J$="T" THEN PRINT AT U,V;"B"
2090 IF J$="C" THEN PRINT AT U,V;"D"
2100 IF J$="A" THEN PRINT AT U,V;"C"
2110 IF J$="Q" THEN PRINT AT U,V;"E"
2120 IF J$="R" THEN PRINT AT U,V;"F"
2130 INPUT "PIEZA COMIDA? (S/N):";K$
2140 IF K$="N" THEN GO TO 2230
```

```
2150 IF K$="S" THEN INPUT "INTRODUCIR CODIG
     O (P/T/C/A/Q/R):";L$
2160 LET BL=BL+1
2170 IF L$="P" THEN PRINT AT 1,7+BL;"H"
2180 IF L$="T" THEN PRINT AT 1,7+BL;"I"
2190 IF L$="C" THEN PRINT AT 1,7+BL;"K"
2200 IF L$="A" THEN PRINT AT 1,7+BL;"J"
2210 IF L$="Q" THEN PRINT AT 1,7+BL;"L"
2220 IF L$="R" THEN PRINT AT 1,7+BL;"M"
2230 INPUT INVERSE 1; FLASH 1;"PROMOCION DE
     PEON? (S/N):";Q$
2240 IF Q$="N" THEN GO TO 1000
2250 IF Q$="S" THEN GO TO 5000
2260 GO TO 1000
```

Había llegado el momento de ejecutar el programa y jugar unos pocos movimientos. No había ningún problema pues todas las piezas movidas eran sustituidas correctamente y las filas de piezas comidas crecían progresivamente en la cabecera y pie de la pantalla. Faltaban añadir las líneas encargadas de la promoción de los peones, que ocuparían la sección 5000. Esta sección hubo de ser dividida en cuatro subsecciones pero fue bastante fácil de programar. Para comenzar, se cambió su primera línea por la línea 5000 REM PROMOCIONES.

```
5000 REM PROMOCIONES
5010 INPUT "PROMOVER A? (T/C/A/Q/R):";T$
5020 IF T$="T" THEN GO TO 5250
5030 IF T$="C" THEN GO TO 5200
5040 IF T$="A" THEN GO TO 5150
5050 IF T$="Q" THEN GO TO 5110
5110 INPUT "REINA A:";X;Y
5120 INPUT "COLOR DE LA REINA? (N/B):";Q$
5130 IF Q$="N" THEN PRINT AT X,Y;"E": GO TO
     1000
5140 IF Q$="B" THEN PRINT AT X,Y;"L": GO TO
     2000
5150 INPUT "ALFIL A:";X;Y
```

```
5160 INPUT "COLOR DE1 ALFIL? (N/B):";Q$
5170 IF Q$="N" THEN PRINT AT X,Y;"C": GO TO
     1000
5180 IF Q$="B" THEN PRINT AT X,Y;"J": GO TO
     2000
5200 INPUT "CABALLO A:";X;Y
5210 INPUT "COLOR DEL CABALLO? (N/B):";Q$
5220 IF Q$="N" THEN PRINT AT X,Y;"D": GO TO
     1000
5230 IF Q$="B" THEN PRINT AT X,Y;"K": GO TO
     2000
5250 INPUT "TORRE A:";X;Y
5260 INPUT "COLOR DE LA TORRE? (N/B):";Q$
5270 IF Q$="N" THEN PRINT AT X,Y;"B": GO TO
     1000
5280 IF Q$="B" THEN PRINT AT X,Y;"I": GO TO
     2000
```

Puede sorprender que se acceda a esta sección mediante GO TO cuando a las secciones 3000 y 3500 se accedía mediante GO SUB. La respuesta está en lo que hace cada sección. En las secciones 3000 y 3500 el ordenador no encuentra lo que busca y vuelve (RETURN), mientras que en la sección 5000 encuentra lo que busca y es enviado a diferentes líneas.

Una vez completada y comprobada esta sección, el programa del juego está terminado y sólo falta eliminar las instrucciones STOP de las líneas 1999, 2999, 3999, 5999 y 7999. Borrar también las líneas 4000 y 4999 si aún no se ha hecho.

A título de simple curiosidad, se averiguó la memoria que utilizaba el programa. Para hacerlo basta escribir (sin ningún número de línea) PRINT PEEK 23627+256*PEEK 23628 – (PEEK 23635+256*PEEK 23636) y luego pulsar ENTER. (El signo * es el símbolo de la multiplicación.) En la parte superior de la pantalla, o donde haya sitio libre, saldrá inmediatamente un número. En este caso salió el 7411. Es el número de bytes de memoria ocupados. Para averiguar

la cantidad de memoria restante, escribir PRINT PEEK 23613+256*PEEK 23614 – (PEEK 23653+256*PEEK 23654) – 100, y pulsar luego ENTER. Salió el 33847. La suma de esos dos números da 41258. Aunque el ordenador utilizado era un Spectrum de 48K, las cifras mostradas ponen bien claramente de manifiesto que este programa de ajedrez entraría perfectamente en un Spectrum de 16K.

8. DAMAS CHINAS

Este juego es para 2 a 6 jugadores, cada uno de los cuales juega con fichas de diferente color. El tablero tiene seis lados y seis vértices y cada color ocupa sólo uno de los seis vértices. De hecho, este juego exige programar cinco juegos distintos en el mismo programa, porque la secuencia de movimientos de cada color es distinta en función del número de jugadores. En consecuencia, el esqueleto de este programa ha de ser de alguna forma diferente a los hechos hasta ahora.

Hay sólo dos caracteres gráficos: una ficha y una cruz (que sustituye a la ficha quitada de una posición). Por lo tanto, la rutina corta de generación de estos dos caracteres gráficos se puede incluir en la sección dedicada a diseñar el tablero, que ubicaremos al final del programa (a partir de la línea 5000). Como hay mucha gente que no conoce el juego de las damas chinas, parece aconsejable presentar al principio una pantalla con las reglas del juego. En este caso le asignaremos una sección ubicada en la primera parte del programa, y a esta sección seguirán otras secciones dedicadas a programar sendas partidas de dos jugadores, de tres jugadores, etc. Esta orientación desemboca en una distribución limpia y fácil de dividir en secciones con objeto de favorecer la introducción de las

líneas de instrucciones y la búsqueda de problemas en caso de que surgiera alguno. El esqueleto es por lo tanto el siguiente:

```
  10 REM * DAMAS CHINAS *
  20 POKE 23609,255: POKE 23658,8
  30 BORDER 0: PAPER 7: CLS
  35 REM * REGLAS *
 150 GOSUB 5000
 200 REM * NO. DE JUGADORES *
 299 STOP
 500 REM * 2 JUGADORES *
 999 REM STOP
1000 REM * 3 JUGADORES *
1499 STOP
1500 REM * 4 JUGADORES *
1999 STOP
2000 REM * 5 JUGADORES *
2499 STOP
2500 REM * 6 JUGADORES *
4999 STOP
5000 REM * DISEÑO DEL TABLERO *
5300 RETURN
```

Este esqueleto constituye una buena guía para escribir las secciones, y está claro que dice que la primera sección que hay que desarrollar son las reglas. Las fichas se mueven a lo largo de las líneas, en cualquier dirección y normalmente sólo un espacio. La única excepción a esta regla es el salto por encima de la ficha de un oponente; sólo se puede saltar en línea recta pero se puede saltar a más de un oponente (como en el juego de damas clásico). No se comen fichas del oponente y el ganador es el que llena primero una esquina opuesta con sus fichas. Ningún jugador puede mantener su ficha en la esquina con la intención de estorbar a un oponente.

Como se verá, las reglas son algo largas y no resultó fácil reducirlas a frases cortas. En cualquier caso, cada uno puede cambiar la explicación siguiente en función de sus preferencias

lingüísticas. Un punto final: esta sección debe incluir también una INPUT que permita borrar la pantalla y presentar el tablero del juego. He aquí el listado:

```
 40 PRINT INVERSE 1; INK 2;AT 1,5;"*REGLAS
    DE LAS DAMAS CHINAS*"
 50 PRINT INK 1;AT 3,0;"*ESTE JUEGO PUEDE
    SER JUGADO";AT 4,1;"POR DOS A SEIS JUG
    ADORES."
 60 PRINT INK 1;AT 5,0;"*EL GANADOR ES EL
    QUE PRIMERO";AT 6,1;"COLOCA UNA FICHA
    EN LA PUNTA";AT 7,1;"OPUESTA DE LA EST
    RELLA"
 70 PRINT INK 1;AT 8,0;"*NADIE PUEDE MANTE
    NER";AT 9,1;"UNA FICHA EN UNA PUNTA PA
    RA";AT 10,1;"QUE NO GANE EL OPONENTE."
 80 PRINT INK 1;AT 11,0;"*LOS JUGADORES MU
    EVEN POR";AT 12,1;"TURNOS MARCADOS POR
    INPUTS)";AT 13,0;"*SE PUEDEN MOVER LAS
    FICHAS"
 90 PRINT INK 1;AT 14,1;"EN CUALQUIER DIRE
    CCION";AT 15,1;"A LO LARGO DE UNA LINE
    A";AT 16,0;"*LAS FICHAS IRAN A AGUJERO
    S";AT 17,1;"ADYACENTES O A UN AGUJERO"
100 PRINT INK 1;AT 18,1;"LIBRE SALTANDO SO
    BRE OTRA.";AT 19,0;"*SE PUEDE DAR MAS
    DE UN SALTO";AT 20,0";"*NO SE COMEN FI
    CHAS."
110 INPUT "Pulsar M para jugar";K$
120 IF K$="M" THEN CLS: GO SUB 5000
```

Las líneas 110 y 120 constituyen la secuencia de cambio de la pantalla de las reglas por la pantalla del juego. Como se ha añadido a la línea 120 la instrucción GO SUBrutina, hay que borrar la línea 150. Para hacerlo, escribir 150 y pulsar ENTER; la línea 150 desaparecerá. Lo mejor será pasar ahora a la sección 5000, dedicada al diseño del tablero.

Para diseñar la figura hexagonal (de seis puntas) de este tablero hubo que hacer muchas pruebas hasta conseguir que la intersección de las líneas coincidiera con el centro de un cuadro de carácter, y para conseguirlo fue indispensable utilizar el tablero de melamina. Al fin se consiguió dibujar una figura que cubría una buena parte de la pantalla; luego se añadieron los números del borde lateral y las letras del borde inferior. Por último se convirtió este diseño en líneas de programa, comenzando por los dos caracteres gráficos necesarios. No olvidar que las A y B encerradas entre comillas deben ser introducidas en el modo gráfico.

Al escribir el programa por primera vez no estaban incluidas las líneas de trazos alineados con los números laterales, pero como todos los números pares apuntaban a una línea comprendida entre dos líneas más obvias correspondientes a números impares, resultaba difícil leer con rapidez la letra y el número correspondientes a un movimento. Por eso se añadieron luego las líneas de trazos alineadas con los números. Mucho cuidado al escribir las líneas 5100 y 5110; hay que contar bien el número de guiones de cada posición. (Los guiones se escriben pulsando las teclas SYMBOL SHIFT y J.) La línea 5130 asigna un número Y a la fila de letras del borde inferior.

```
5010 FOR X=0 TO 1: FOR Y=0 TO 7: READ Z: PO
     KE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
5020 DATA 60,126,255,255,255,255,126,60,8,1
     37,106,28,60,203,8,8: REM A="A",B="B"
5030 PLOT 36,80: DRAW 0,24: PLOT 52,72: DRA
     W 0,40: PLOT 68,16: DRAW 0,152: PLOT 8
     4,24: DRAW 0,136: PLOT 100,32: DRAW 0,
     120: PLOT 116,40: DRAW 0,104
5040 PLOT 132,32: DRAW 0,120: PLOT 148,24:
     DRAW 0,136: PLOT 164,16: DRAW 0,152: P
     LOT 180,72: DRAW 0,40: PLOT 196,80: DR
     AW 0,24
```

```
5050 PLOT 64,146: DRAW 24,12: PLOT 64,130:
     DRAW 40,20: PLOT 16,90: DRAW 152,76: P
     LOT 32,82: DRAW 136,68: PLOT 48,74: DR
     AW 120,60: PLOT 64,66: DRAW 104,52
5060 PLOT 64,50: DRAW 120,60: PLOT 64,34: D
     RAW 136,68: PLOT 64,18: DRAW 152,76: P
     LOT 128,34: DRAW 40,20: PLOT 114,26: D
     RAW 24,12
5070 PLOT 144,158: DRAW 24,-12: PLOT 128,15
     0: DRAW 40,-20: PLOT 64,166: DRAW 152,
     -76: PLOT 64,150: DRAW 136,-68: PLOT 6
     4,134: DRAW 120,-60: PLOT 64,118: DRAW
     104,-52
5080 PLOT 48,110: DRAW 120,-60: PLOT 32,102
     : DRAW 136,-68: PLOT 16,94: DRAW 152,-
     76: PLOT 64,54: DRAW 40,-20: PLOT 64,3
     8: DRAW 24,-12
5090 PRINT INVERSE 1; INK 1;AT 0,0;"DAMAS C
     HINAS"
5100 PRINT AT 1,1;"1-----";AT 2,1;"2------"
     ;AT 3,1;"3-----";AT 4,1;"4------";AT 5
     ,1;"5-----";AT6,1;"6------";AT 7,1;"7-
     ----";AT 8,1;"8---"
5110 PRINT AT 9,1;"9-";AT 10,0;"10";AT 11,0
     ;"11-";AT 12,0;"12---";AT 13,0;"13----
     -";AT 14,0;"14------";AT 15,0;"15-----
     ";AT 16,0;"16------";AT 17,0;"17-----"
     ;AT 18,0;"18------";AT 19,0;"19-----"
5120 PRINT AT 20,2;"A B C D E F G H I J K L
     M"
5130 LET A=2: LET B=4: LET C=6: LET D=8: LE
     T E=10: LET F=12: LET G=14: LET H=16:
     LET I=18: LET J=20: LET K=22: LET L=24
     : LET M=26
5200 PRINT INK 1;AT 14,23;"TECLEAR";AT 15,2
     3;"PRIMERO";AT 16,23;"NUMERO.";AT 17,2
     3;"DESPUES";AT 18,23;"LETRA."
5300 RETURN
```

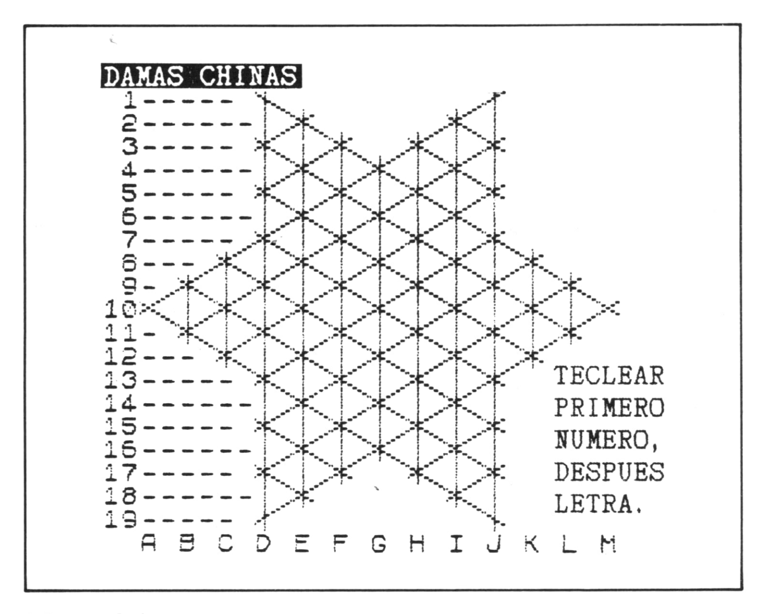


Figura 8.1

Una vez introducidas estas líneas, ejecutar el programa; el resultado será la presentación en pantalla de la figura 8.1. La razón de haber incluido las líneas de trazos alineados con los números queda ahora bien clara: sin su guía no resultaría nada fácil localizar el punto 4E, por ejemplo.

Este tablero no tiene ninguna ficha porque el programa necesita una INPUT que pregunte el número de jugadores para que pueda actuar en consecuencia, a cuyo efecto hay que escribir ahora la sección 200. Es corta pues sólo se encarga de dirigir al ordenador a una de las otras cinco secciones en función del número de jugadores.

```
210 INPUT "CUANTOS JUGADORES? (2/3/4/5/6):
    ";J$
220 IF J$="2" THEN GO TO 500
230 IF J$="3" THEN GO TO 1000
```

```
240 IF J$="4" THEN GO TO 1500
250 IF J$="5" THEN GO TO 2000
260 IF J$="6" THEN GO TO 2500
```

Ahora se puede pasar a escribir la primera de las cinco secciones, que programa la partida de dos jugadores (no olvidar que constituirá un modelo para las demás secciones, más largas). Hay que poner las fichas en sus posiciones, hay que imprimir en el espacio libre de la esquina superior derecha de la pantalla el número de jugadores y el orden de salida, y hay que asignar turnos a los jugadores mediante dos INPUT. Después irán las instrucciones que digan al ordenador que imprima el carácter gráfico B, una cruz, en el cruce de donde se quite una ficha (en sustitución de las líneas del tablero borradas), y el carácter gráfico A, una ficha del color correspondiente, en la nueva posición. Una vez programado todo esto para el primer jugador, hay que repetirlo todo para el otro (con una ficha de diferente color). Luego el ordenador vuelve a la línea 550 para repetir los movimientos las veces que haga falta.

```
520 PRINT INK 3;AT 8,22;"A";AT 9,24;"A";AT
    10,22;"A";AT 10,26;"A";AT 11,24;"A";AT
    12,22;"A"
530 PRINT INK 6;AT 8,6;"A";AT 9,4;"A";AT 1
    0,2;"A";AT 10,6;"A";AT 11,4;"A";AT 12,
    6;"A"
540 PRINT AT 1,22;"2 JUGADO.";AT 2,22;"AMA
    RILLAS";AT 3,22;"MAGENTAS;AT 4,22;"JUG
    AR";AT 5,22;"EN ESE";AT 6,22;"ORDEN"
550 INPUT "AMARILLA DE (NO. Y LETRA):";U;V
560 INPUT "AMARILLA A (NO. Y LETRA):";X;Y
570 PRINT INK 0;AT U,V;"B"
580 PRINT INK 6;AT X,Y;"A"
590 INPUT "MAGENTA DE (NO. Y LETRA):";U;V
600 INPUT "MAGENTA A (NO. Y LETRA):";X;Y
620 PRINT INK 0;AT U,V;"B"
630 PRINT INK 3;AT X,Y;"A"
640 GO TO 550
```

En las líneas 520, 530, 580 y 630, el carácter gráfico es "A", la ficha, y en las líneas 570 y 620 es "B", la cruz. Se habrá observado que las variables PRINT AT asignadas a las INPUT "DE" son U,V, mientras que las asignadas a las INPUT "A" son X,Y. Anteriormente, en el final de la sección 5000, asignamos las letras A a la M a los números Y del tablero. Estas letras están ahora en la memoria y si se utilizaran para otros fines podrían confundir al ordenador pues las variables de este tipo sólo se deben utilizar con un mismo fin a lo largo de todo el programa. Las letras asociadas a los caracteres gráficos están en otra parte de la memoria del odenador y por eso no crean confusiones. No siempre se entiende bien que en la memoria del ordenador hay, además del alfabeto normal A a la Z (mayúsculas y minúsculas), un conjunto extra de letras, A a la U, que es el que utilizan los caracteres gráficos. Estos caracteres gráficos sólo son accesibles en el modo gráfico.

Ahora que todo está ya dispuesto para que jueguen dos jugadores, convendría hacer una prueba antes de seguir adelante. Hacerla y repasar todas las entradas en caso de que el programa no vaya bien. Queremos subrayar la conveniencia de hacer esa prueba porque todo el mundo puede cometer errores al introducir líneas de programa. Concretamente, cuando introdujimos esta sección e hicimos la prueba, tras los dos primeros movimientos, cuando el ordenador debía dar turno al segundo movimiento de las amarillas, no lo hizo así sino que volvió a imprimir las fichas amarilla y magenta que los jugadores acababan de mover; había entonces siete fichas en vez de seis. En seguida descubrimos la posible causa: la GO TO de la línea 640 no era GO TO 550 sino GO TO 520. Un pequeño resbalón muy fácil de cometer. En consecuencia, repasar una y otra vez el programa si no hiciera lo que tiene que hacer.

Una vez introducida la sección dedicada a la partida de 2 jugadores, es el momento de pasar a la sección 1000 (3 jugadores). Es similar en su estructura a la sección 500, pero

hay algunas diferencias en los colores y un conjunto de movimientos más.

```
1020 PRINT INK 6;AT 8,6;"A";AT 9,4;"A";AT 1
     0,2;"A";AT 10,6;"A";AT 11,4;"A";AT 12,
     6;"A"
1030 PRINT INK 2;AT 1,20;"A";AT 2,18;"A";AT
     3,16;"A";AT 3,20;"A";AT 4,18;"A";AT 5,
     20;"A"
1040 PRINT INK 4;AT 15,20;"A";AT 16,18;"A";
     AT 17,16;"A";AT 17,20;"A";AT 18,18;"A"
     ;AT 19,20;"A"
1050 PRINT AT 1,22;"3 JUGADO.";AT 2,22;"AMA
     RILLAS";AT 3,22;"ROJAS Y";AT 4,22;"VER
     DES";AT 5,22"JUGAR";AT 6,22;"EN ESE";A
     T 7,24;"ORDEN"
1060 INPUT "AMARILLA DE (NO. Y LETRA):";U;V
1070 INPUT "AMARILLA A (NO. Y LETRA):";X;Y
1080 PRINT INK 0;AT U,V;"B"
1090 PRINT INK 6;AT X,Y;"A"
1100 INPUT "ROJA DE (NO. Y LETRA):";U;V
1110 INPUT "ROJA A (NO. Y LETRA):";X;Y
1120 PRINT INK 0;AT U,V;"B"
1130 PRINT INK 2;AT X,Y;"A"
1140 INPUT "VERDE DE (NO. Y LETRA):";U;V
1150 INPUT "VERDE A (NO. Y LETRA):";X;Y
1160 PRINT INK 0;AT U,V;"B"
1170 PRINT INK 4;AT X,Y;"A"
1200 GO TO 1060
```

Los colores utilizados esta vez son amarillo, rojo y verde. Como la estrella es de seis puntas, estas fichas no están perfectamente opuestas entre sí (hay dos fichas en el lado izquierdo y una en el lado derecho), por lo que al principio del juego cada jugador debe declarar cuál de los otros dos es su enemigo. Lo normal es elegir al siguiente en el sentido de giro de las agujas del reloj. Una vez introducidas esas líneas, jugar

una partida con tres jugadores a título de prueba, para comprobar que no hay problemas.

```
1510 PRINT INK 1;AT 1,8;"A";AT 2,10;"A";AT
     3,8;"A";AT 3,12;"A";AT 4,10;"A";AT 5,8
     ;"A"
1520 PRINT INK 2;AT 1,20;"A";AT 2,18;"A";AT
     3,16;"A";AT 3,20;"A";AT 4,18;"A";AT 5,
     20;"A"
1530 PRINT INK 5;AT 15,8;"A";AT 16,10;"A";A
     T 17,8;"A";AT 17,12;"A";AT 18,10;"A";A
     T 19,8;"A"
1540 PRINT INK 4;AT 15,20;"A";AT 16,18;"A";
     AT 17,16;"A";AT 17,20;"A";AT 18,18;"A"
     ;AT 19,20;"A"
1550 PRINT AT 1,22;"4 JUGADO.";AT 2,22;"AZU
     LES,RO";AT 3,22;"JAS,CIAN";AT 4,22;"Y
     VERDES";AT 5,22"JUGAR";AT 6,22;"EN ESE
     ";AT 7,24;"ORDEN"
1560 INPUT "AZUL DE (NO. Y LETRA):";U;V
1570 INPUT "AZUL A (NO. Y LETRA):";X;Y
1580 PRINT INK 0;AT U,V;"B"
1590 PRINT INK 1;AT X,Y;"A"
1600 INPUT "ROJA DE (NO. Y LETRA):";U;V
1610 INPUT "ROJA A (NO. Y LETRA):";X;Y
1620 PRINT INK 0;AT U,V;"B"
1630 PRINT INK 2;AT X,Y;"A"
1640 INPUT "CIAN DE (NO. Y LETRA):";U;V
1650 INPUT "CIAN A (NO. Y LETRA):";X;Y
1660 PRINT INK 0;AT U,V;"B"
1670 PRINT INK 5;AT X,Y;"A"
1680 INPUT "VERDE DE (NO. Y LETRA):";U;V
1690 INPUT "VERDE A (NO. Y LETRA):";X;Y
1700 PRINT INK 0;AT U,V;"B"
1710 PRINT INK 4;AT X,Y;"A"
1720 GO TO 1560
```

Esta es la sección 1500 dedicada a programar la partida de

4 jugadores: azules, rojas, verdes y cian. Comprobar otra vez que no hay ningún problema. Los listados se alargan cada vez más e introducirlos puede ser una labor ciertamente aburrida, pero la constancia es una gran virtud. Ahora vienen los dos listados finales.

```
2010 PRINT INK 6;AT 8,6;"A";AT 9,4;"A";AT 1
     0,2;"A";AT 10,6;"A";AT 11,4;"A";AT 12,
     6;"A"
2020 PRINT INK 1;AT 1,8;"A";AT 2,10;"A";AT
     3,8;"A";AT 3,12;"A";AT 4,10;"A";AT 5,8
     ;"A"
2030 PRINT INK 2;AT 1,20;"A";AT 2,18;"A";AT
     3,16;"A";AT 3,20;"A";AT 4,18;"A";AT 5,
     20;"A"
2040 PRINT INK 3;AT 8,22;"A";AT 9,24;"A";AT
     10,22;"A";AT 10,26;"A";AT 11,24;"A";AT
     12,22;"A"
2050 PRINT INK 4;AT 15,20;"A";AT 16,18;"A";
     AT 17,16;"A";AT 17,20;"A";AT 18,18;"A"
     ;AT 19,20;"A"
2060 PRINT AT 1,22;"5 JUGADO.";AT 2,22;"AMA
     RILLAS";AT 3,22;"AZULES,RO";AT 4,22;"J
     AS,MAGEN";AT 5,22;"TA,VERDES";AT 6,22"
     JUGAR";AT 7,24;"EN ESE";AT 8,26;"ORDEN
     "
2070 INPUT "AMARILLA DE (NO. Y LETRA):";U;V
2080 INPUT "AMARILLA A (NO. Y LETRA):";X;Y
2090 PRINT INK 0;AT U,V;"B"
2100 PRINT INK 6;AT X,Y;"A"
2110 INPUT "AZUL DE (NO. Y LETRA):";U;V
2120 INPUT "AZUL A (NO. Y LETRA):";X;Y
2130 PRINT INK 0;AT U,V;"B"
2140 PRINT INK 1;AT X,Y;"A"
2150 INPUT "ROJA DE (NO. Y LETRA):";U;V
2160 INPUT "ROJA A (NO. Y LETRA):";X;Y
2170 PRINT INK 0;AT U,V;"B"
2180 PRINT INK 2;AT X,Y;"A"
```

```
2190 INPUT "MAGENTA DE (NO. Y LETRA):";U;V
2200 INPUT "MAGENTA A (NO. Y LETRA):";X;Y
2210 PRINT INK 0;AT U,V;"B"
2220 PRINT INK 3;AT X,Y;"A"
2230 INPUT "VERDE DE (NO. Y LETRA):";U;V
2240 INPUT "VERDE A (NO. Y LETRA):";X;Y
2250 PRINT INK 0;AT U,V;"B"
2260 PRINT INK 4;AT X,Y;"A"
2270 GO TO 2070
```

La sección 2000 programa una partida de 5 jugadores. También ahora es necesario tomar una decisión con respecto al color enemigo de cada jugador. Lo normal es que lo sea el segundo oponente en el sentido de giro de las agujas del reloj.

```
2510 PRINT INK 1;AT 1,8;"A";AT 2,10;"A";AT
     3,8;"A";AT 3,12;"A";AT 4,10;"A";AT 5,8
     ;"A"
2520 PRINT INK 2;AT 1,20;"A";AT 2,18;"A";AT
     3,16;"A";AT 3,20;"A";AT 4,18;"A";AT 5,
     20;"A"
2530 PRINT INK 3;AT 8,22;"A";AT 9,24;"A";AT
     10,22;"A";AT 10,26;"A";AT 11,24;"A";AT
     12,22;"A"
2540 PRINT INK 4;AT 15,20;"A";AT 16,18;"A";
     AT 17,16;"A";AT 17,20;"A";AT 18,18;"A"
     ;AT 19,20;"A"
2550 PRINT INK 5;AT 15,8;"A";AT 16,10;"A";A
     T 17,8;"A";AT 17,12;"A";AT 18,10;"A";A
     T 19,8;"A"
2560 PRINT INK 6;AT 8,6;"A";AT 9,4;"A";AT 1
     0,2;"A";AT 10,6;"A";AT 11,4;"A";AT 12,
     6;"A"
2570 PRINT AT 0,22;"6 JUGADO.";AT 1,22;"ROJ
     AS,AZU";AT 2,22;"LES,MAGEN";AT 3,22;"T
     A,VERDES";AT 4,22;"CIAN Y";AT 5,22;"AM
     ARILLAS";AT 6,22;"JUGAR";AT 7,24;"EN E
     SE";AT 8,26;"ORDEN"
```

```
2580 INPUT "AZUL DE (NO. Y LETRA):";U;V
2590 INPUT "AZUL A (NO. Y LETRA):";X;Y
2600 PRINT INK 0;AT U,V;"B"
2610 PRINT INK 1;AT X,Y;"A"
2620 INPUT "ROJA DE (NO. Y LETRA):";U;V
2630 INPUT "ROJA A (NO. Y LETRA):";X;Y
2640 PRINT INK 0;AT U,V;"B"
2650 PRINT INK 2;AT X,Y;"A"
2660 INPUT "MAGENTA DE (NO. Y LETRA):";U;V
2670 INPUT "MAGENTA A (NO. Y LETRA):";X;Y
2680 PRINT INK 0;AT U,V;"B"
2690 PRINT INK 3;AT X,Y;"A"
2700 INPUT "VERDE DE (NO. Y LETRA):";U;V
2710 INPUT "VERDE A (NO. Y LETRA):";X;Y
2720 PRINT INK 0;AT U,V;"B"
2730 PRINT INK 4;AT X,Y;"A"
2740 INPUT "CIAN DE (NO. Y LETRA):";U;V
2750 INPUT "CIAN A (NO. Y LETRA):";X;Y
2760 PRINT INK 0;AT U,V;"B"
2770 PRINT INK 5;AT X,Y;"A"
2780 INPUT "AMARILLA DE (NO. Y LETRA):";U;V
2790 INPUT "AMARILLA A (NO. Y LETRA):";X;Y
2800 PRINT INK 0;AT U,V;"B"
2810 PRINT INK 6;AT X,Y;"A"
2820 GO TO 2580
```

Esta es la última sección, la que programa la partida de 6 jugadores. Una vez probada se pueden borrar todas las líneas STOP. En su forma actual, el programa completo es muy largo y por eso precisamente merece la pena que el lector intente acortarlo un poco a título de ejercicio. Un sitio muy claro es el diseño del tablero de seis lados; en las líneas 5030 a 5080 hay 33 instrucciones PLOT y DRAW que hacen perfectamente lo que tienen que hacer, pero esta rutina se puede reducir a la mitad incorporando un bucle FOR/NEXT.

Esta es una buena demostración de las muchas formas de empleo del bucle FOR/NEXT. A fin de tener un ejemplo in-

dependiente, haremos un pequeño programa. Como habrá que introducirlo en un ordenador limpio de otros datos, guardar en cinta todo lo hecho hasta ahora grabándolo y verificándolo adecuadamente; después desconectar y conectar el ordenador para borrar todos los caracteres gráficos y todas las variables. Introducir las líneas siguientes:

```
10 REM PRUEBA
20 FOR X=1 TO 33: READ A,B,C,D: PLOT A,B:
   DRAW C,D: NEXT X
30 DATA 36,80,0,24,52,72,0,40,68,16,0,152
   ,84,24,0,136,100,32,0,120,116,40,0,104
   ,132,32,0,120,148,24,0,136,164,16,0,15
   2,180,72,0,40,196,80,0,24
40 DATA 64,146,24,12,64,130,40,20,16,90,1
   52,76,32,82,136,68,48,74,120,60,64,66,
   104,52,64,50,120,60,64,34,136,68,64,18
   ,152,76,128,34,40,20,144,26,24,12
50 DATA 144,158,24,-12,128,150,40,-20,64,
   166,152,-76,64,150,136,-68,64,134,120,
   -60,64,118,104,-52,48,110,120,-60,32,1
   02,136,-68,16,94,152,-76,64,54,40,-20,
   64,38,24,-12
```

Una vez introducido y ejecutado este programa, sale en la pantalla el conjunto de líneas que componen el tablero. Las cuatro líneas cortas de este nuevo programa hacen lo mismo que las seis líneas largas (5030 a 5080) del programa anterior. La explicación es ésta:

Hay 33 instrucciones PLOT y DRAW independientes. La línea 20 se encarga de ello: 20 FOR X=1 TO 33 anticipa al ordenador que tendrá que hacer algo 33 veces y asigna a X el número de veces que lo ha hecho; READ A,B,C,D dice al ordenador que lea los números DATA siguientes en lotes de cuatro; PLOT A,B manda considerar los dos primeros números del lote de cuatro como números PLOT; DRAW C,D

manda considerar los dos segundos números del lote de cuatro como números DRAW; NEXT X dice al ordenador que vuelva atrás para leer el segundo lote de 4 números y repetir el proceso, y así hasta 33 veces.

Las tres líneas de números DATA son por lo tanto los números PLOT y DRAW de las líneas originales 5030 y 5080, introducidos en la misma secuencia y con una coma entre cada dos números consecutivos.

Es posible que no merezca la pena tomarse la molestia de transferir estas líneas al programa escrito anteriormente, pero si alguien quiere hacerlo, que cambie los números de línea de este programa por 5025, 5045, 5055 y 5065, que lo grabe después en cinta (con SAVE) y que limpie el ordenador; después tendrá que cargar el programa DAMAS CHINAS (con LOAD), listarlo (con LIST) para ver si todo está bien, borrar las líneas 5030 a 5080, y luego mezclar (con MERGE) este pequeño programa con el anterior. Para hacer esto último basta poner en la unidad de cintas la cinta que contiene el programa pequeño, comprobar que el programa grande está todavía en la memoria del ordenador y luego escribir MERGE “ ”. A la recepción de esta orden, el ordenador mezclará el programa pequeño con el grande y lo ejecutará como antes, pero ahora hay once líneas de instrucciones menos.

Hay una solución para reducir a unas pocas un gran número de líneas de cada una de las secciones de juegos independientes (2 a 6 juugadores), pero para ello hay que poner en juego una serie de complicados razonamientos y además hay que trabajar con variables de cadena, por lo que será mejor dejarlo para cuando se domine mucho más la programación en BASIC.

Lo más importante a estas alturas es desarrollar la capacidad de discernir qué se quiere que haga un programa y saber levantar a partir de ahí un esqueleto o estructura compuesta

por varias secciones que sean fáciles escribir. Si realmente se quiere aprender a escribir programas, la *estructura* es lo primero a tener en cuenta. El método adoptado en este libro lleva al lector por el buen camino.

9. FORMA DE EMPLEO DE PLOT Y DRAW

En los capítulos anteriores han salido muchos ejemplos de generación de caracteres gráficos y diferentes formas de empleo de los bucles FOR/NEXT, algunas prácticas con INPUT y algunos ejercicios con variables, e instrucciones PRINT AT. También han salido a relucir las instrucciones PLOT y DRAW, pero en menor grado. Ahora es el momento de ampliar nuestros conocimientos sobre este punto.

PLOT y DRAW trabajan con las coordenadas "PIXEL": de izquierda a derecha a lo largo de la parte inferior de la pantalla están numeradas del 0 al 255 (columnas), y de abajo arriba en el lado derecho están numeradas del 0 al 176 (filas). Por ejemplo: PLOT 120,86 es un punto cercano al centro de la pantalla. No olvidar que las coordenadas PIXEL se escriben siempre en este orden: coordenada lateral (fila), coordenada superior (columna). Grabemos esto en la memoria.

A la hora de escribir programas, no son pocas las veces que conviene incluir una figura, por ejemplo un triángulo, para conseguir el buen efecto necesario. Pero no siempre es fácil recordar la forma exacta de definir un triángulo que señale una u otra dirección, y por eso sería de gran utilidad tener a mano un programa que trace triángulos de diferentes for-

mas. El siguiente programa, escrito con este fin, dibuja en la pantalla triángulos e imprime debajo de cada uno el número de su línea; de esta forma será fácil localizarlos en el futuro a simple vista. El programa estará organizado en secciones. He aquí la primera parte:

```
10 REM TRIANGULOS
20 POKE 23658,8: POKE 23609,255: BORDER 1
   : PAPER 7: INK 0: CLS
30 FOR X=10 TO 0 STEP -2: PLOT 14-X,X-172
   : DRAW X,0: NEXT X
40 PRINT AT 2,0;"30"
```

Estas líneas dan título al programa, hacen que la pulsación de las teclas vaya acompañada por un pitido (BEEP), ponen el ordenador en modo CAPS (mayúsculas), proporcionan un borde azul y, por la acción de una sola línea (30), dibujan un triángulo en la esquina superior izquierda de la pantalla. La línea 40 imprime su número.

Obsérvese el STEP –2; si se hubiera escrito STEP –1, el triángulo habría sido lleno. Obsérvese también el empleo de "—X,X–" en la instrucción PLOT pues es importante. El listado siguiente llena esa línea superior de la pantalla con triángulos que apuntan a diversas direcciones. Un triángulo será negro.

```
 50 FOR X=0 TO 10: PLOT 26,172-X: DRAW X,0
    :NEXT X
 60 PRINT AT 2,3;"50"
 70 FOR X=10 TO 0 STEP -2: PLOT 60,162+X:
    DRAW -X,0: NEXT X
 80 PRINT AT 2,6;"70"
 90 FOR X=10 TO 0 STEP -2: PLOT 76,162+X:
    DRAW X,0: NEXT X
100 PRINT AT 2,9;"90"
110 FOR X=10 TO 0 STEP -2: PLOT X+102,162:
    DRAW 0,X: NEXT X
```

```
120 PRINT AT 2,12;"110"
130 FOR X=0 TO 10 STEP 2: PLOT 144-X,162:
    DRAW 0,X: NEXT X
140 PRINT AT 2,16;"130"
150 FOR X=10 TO 0 STEP -2: PLOT X+616,172:
    DRAW 0,-X: NEXT X
160 PRINT AT 2,20;"150"
170 FOR X=10 TO 0 STEP -2: PLOT X-204,172:
    DRAW 0,-X: NEXT X
180 PRINT AT 2,24;"170"
190 FOR X=40 TO 0 STEP -2: PLOT X+210,162:
    DRAW 0,X*.25: NEXT X
200 PRINT AT 2,28;"190"
```

En la línea 50, como los números FOR X aumentan (van del 0 al 10, no del 10 al 0), sólo hay necesidad de poner STEP si el salto es distinto de 1. La cuenta atrás de los números exige incluir STEP, incluso si éste es 1. El paso 1 hace que la figura sea llena, no definida por líneas. A continuación de STEP se puede poner cualquier número; el ordenador obedecerá siempre.

Al ejecutar las líneas anteriores se verá que los ejemplos 30 y 110 tienen la misma forma pero sus líneas apuntan a diferentes direcciones, lo cual puede tener sus aplicaciones. La diferencia está en la instrucción DRAW: la línea 30 tiene DRAW X,0 y la línea 110 tiene DRAW 0,X. Cuando la X está en el primer lugar las líneas son horizontales y cuando está en el segundo lugar son verticales.

Estudiar detenidamente estos ejemplos comparando las diferencias entre triángulos de la misma o similar forma. Hay que tener mucho cuidado con los signos más (+) y menos (–) de las instrucciones PLOT y DRAW pues tienen importantes efectos. Por eso merece la pena dedicar un tiempo a cualquiera de ellas para analizar cómo se las arregla para conseguir el efecto deseado y descubrir lo que pasaría si se cambiara ese signo. Obsérvese la diferencia entre las líneas 110 y 190.

El siguiente listado, que llena de triángulos la segunda fila de la pantalla, traza triángulos de distinta forma. Introducir ahora estas líneas.

```
210 FOR X=10 TO 0 STEP -2: PLOT 14-X,X-144
    : DRAW X*2,0: NEXT X
220 PRINT AT 6,0;"210"
230 FOR X=0 TO 10 STEP 2: PLOT 44-X,X+134:
    DRAW X*2,0: NEXT X
240 PRINT AT 6,4;"230"
250 FOR X=10 TO 0 STEP -2: PLOT 82-X,X-140
    : DRAW 0,X*2: NEXT X
260 PRINT AT 6,8;"250"
270 FOR X=0 TO 10 STEP 2: PLOT 102+X,X-140
    : DRAW 0,+X*2: NEXT X
280 PRINT AT 6,12;"270"
290 FOR X=0 TO 20 STEP 2: PLOT 140-X/3,X-1
    50: DRAW X/3*2,0: NEXT X
300 PRINT AT 6,16;"290"
310 FOR X=20 TO 0 STEP -2: PLOT 172-X/3,10
    3+X: DRAW X/3*2,0: NEXT X
320 PRINT AT 6,20;"310"
330 FOR X=0 TO 20 STEP 2: PLOT 214-X,140+X
    /3: DRAW 0,-X/3*2: NEXT X
340 PRINT AT 6,24;"330"
350 FOR X=20 TO 0 STEP -2: PLOT 224+X,140-
    X/3: DRAW 0,X/3*2: NEXT X
360 PRINT AT 6,28;"350"
```

Si se ejecuta ahora el programa, saldrán en la pantalla dos filas de triángulos surtidos. Llamar a una línea con LIST y su número y EDITarla luego (CAPS SHIFT y 1); bajará al pie de la pantalla en disposición de ser modificada. En primer lugar, cambiar su número por el 1000 con objeto de que no interfiera con ninguna línea introducida hasta ahora. Cuando haya sido modificada y mandada de nuevo al programa, RUN 1000 presentará en la pantalla únicamente el resultado de la línea modificada.

Hacer modificaciones en únicamente un punto cada vez, ya sea un número o un signo, introducir RUN 1000 y estudiar el resultado de la modificación. Una o dos horas dedicadas a "jugar" con modificaciones de este tipo enseñarán más programación que mil palabras. Cuando se haya acabado, se puede borrar la línea simplemente escribiendo su número de línea y pulsando luego ENTER. Esa línea se desvanecerá y la línea de la que se había sacado seguirá ahí igual que al principio.

La mayoría de los triángulos trazados hasta ahora eran bastante regulares, pero ahora viene una fila de triángulos de forma diferente.

```
370 FOR X=0 TO 10 STEP 2: PLOT 6+X,X+96: D
    RAW 0,X*1.25: NEXT X
380 PRINT AT 10,0;"370"
390 FOR X=10 TO 0 STEP -2: PLOT 48-X,X+96:
    DRAW 0,X*1.25: NEXT X
400 PRINT AT 10,4;"390"
410 FOR X=20 TO 0 STEP -2: PLOT X-86,X-118
    : DRAW 0,X*.5: NEXT X
420 PRINT AT 10,8;"410"
430 FOR X=20 TO 0 STEP -2: PLOT 97+X,X-118
    : DRAW 0,X*.5: NEXT X
440 PRINT AT 10,12;"430"
450 FOR X=0 TO 10 STEP 2: PLOT 124+X,X-112
    : DRAW X*2,0: NEXT X
460 PRINT AT 10,16;"450"
470 FOR X=10 TO 0 STEP -2: PLOT 187-X,X-11
    2: DRAW -X*2,0: NEXT X
480 PRINT AT 10,20;"470"
490 FOR X=10 TO 0 STEP -2: PLOT X+188,X+10
    0: DRAW X*2,0: NEXT X
500 PRINT AT 10,24;"490"
510 FOR X=10 TO 0 STEP -2: PLOT 252-X,X+10
    0: DRAW -X*2,0: NEXT X
520 PRINT AT 10,28;"510"
```

Estos triángulos se diferencian en que están inclinados en distintas direcciones. Un detenido estudio comparativo de las instrucciones PLOT de las líneas 370 a 430 pondrá de manifiesto que las diferencias tienen su origen en los signos + o – de la instrucción PLOT. Más adelante, en las líneas 450 a 510, se consigue un efecto absolutamente distinto modificando la instrucción DRAW para que trace líneas horizontales y jugando con los signos + o – de la instrucción PLOT.

```
530 FOR X=0 TO 10 STEP 2: PLOT 2,X+64: DRA
    W X*2,0: NEXT X
540 PRINT AT 15,0;"530"
550 FOR X=10 TO 0 STEP -2: PLOT 54,X+64: D
    RAW -X*2,0: NEXT X
560 PRINT AT 15,4;"550"
570 FOR X=0 TO 10 STEP 2: PLOT X-80,60: DR
    AW 0,X*2: NEXT X
580 PRINT AT 15,8;"570"
590 FOR X=0 TO 10 STEP 2: PLOT X+102,60: D
    RAW 0,X*2: NEXT X
600 PRINT AT 15,12;"590"
610 FOR X=0 TO 10 STEP 2: PLOT 144-X,62: D
    RAW X,0+X*2: NEXT X
620 PRINT AT 15,16;"610"
630 FOR X=0 TO 10 STEP 2: PLOT 160+X,63: D
    RAW -X,0+X*2: NEXT X
640 PRINT AT 15,20;"630"
650 FOR X=0 TO 10 STEP 2: PLOT 208-X,62: D
    RAW -X,0+X*2: NEXT X
660 PRINT AT 15,24;"650"
670 FOR X=0 TO 10 STEP 2: PLOT 228+X,62: D
    RAW X,0+X*2: NEXT X
680 PRINT AT 15,28;"670"
690 FOR X=0 TO 10 STEP 2: PLOT 18-X,40: DR
    AW X,0-X*2: NEXT X
700 PRINT AT 20,0;"690"
710 FOR X=0 TO 10 STEP 2: PLOT 42+X,40: DR
    AW -X,0-X*2: NEXT X
```

```
720 PRINT AT 20,4;"710"
730 FOR X=0 TO 10 STEP 2: PLOT 86-X,40: DR
    AW -X,0-X*2: NEXT X
740 PRINT AT 20,8;"730"
750 FOR X=0 TO 10 STEP 2: PLOT 98+X,40: DR
    AW X,0-X*2: NEXT X
760 PRINT AT 20,12;"750"
770 FOR X=0 TO 10 STEP 2: PLOT 138-X,20: D
    RAW X*2,0+X*2: NEXT X
780 PRINT AT 20,16;"770"
790 FOR X=0 TO 10 STEP 2: PLOT X+170,20: D
    RAW -X*2,0+X*2: NEXT X
800 PRINT AT 20,20;"790"
810 FOR X=0 TO 10 STEP 2: PLOT 206-X,16+X:
    DRAW X,0+X*2: NEXT X
820 PRINT AT 20,24;"810"
830 FOR X=10 TO 0 STEP -2: PLOT 230+X,16+X
    : DRAW -X,X*2: NEXT X
840 PRINT AT 20,28;"830"
```

Estas dos líneas finales de triángulos muestran algunos efectos más, que se pueden conseguir jugando con los signos + y – de las instrucciones PLOT y DRAW. Cualquiera de estos triángulos puede ser de gran utilidad a la hora de escribir un programa, pero uno de ellos èn particular llama la atención. Se trata del ejemplo 790, que recuerda la vela de un bajel árabe.

Como merece la pena seguir esta idea, añadiremos algo más al programa. La pantalla está llena ya, por lo que habrá que pasar a una segunda pantalla. Esta vez hemos seguido un método diferente con objeto de mostrar una alternativa a la orden INPUT.

Hay una tecla (la tecla N de la fila inferior) que lleva grabada en verde la palabra INKEY$. Se accede a ella pulsando simultáneamente CAPS SHIFT y SYMBOL SHIFT y soltándolas luego; esta acción pone al ordenador en el modo “E” y

en este modo la pulsación de la tecla N pone en acción la orden INKEY$.

```
 850 PRINT INK 1;"PULSAR P PARA CONTINUAR":
     PAUSE 10000: IF INKEY$="P" OR INKEY$="
     p" THEN CLS; GO TO 1000
1000 PRINT AT 10,10;"A"
```

La palabra "PRINT" sin más hace que el ordenador se ponga a imprimir en la línea siguiente a la que acaba de imprimir; en este caso se trata de la línea inferior de la pantalla. Se ha incluido INK 1 (azul) para que resalte más. Primero hay una instrucción INPUT que pide pulsar la tecla P, y a continuación hay dos instrucciones IF/THEN combinadas en una: P mayúscula y p minúscula. Estas instrucciones dicen al ordenador que si es pulsada una de estas teclas debe limpiar la pantalla (CLS) e ir después a la línea 1000 (GO TO 1000), que es precisamente donde estará el siguiente programa. Se ha incluido la instrucción PAUSE 10000 para mantener al ordenador en el modo adecuado para recibir la orden P; dura exactamente 3 minutos pues una PAUSE dura 1/50 segundos. La línea 1000 es una línea temporal que ha sido incluida únicamente para mostrar que la orden INKEY$ está en activo. Ejecutar ahora el programa: tras pulsar la P, saldrá una A en las cercanías del centro de la pantalla.

Ahora se puede pasar a escribir el listado de la línea 1000 en adelante. Un bajel árabe trae reminiscencias del Nilo, luego habrá que añadir algunos caracteres gráficos para representar algo de agua y cielo. Vaya primero el fondo.

```
1000 REM ESCENA DEL NILO
1010 FOR X=0 TO 1: FOR Y=0 TO 7: READ Z: PO
     KE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
1020 DATA 0,0,60,195,0,0,60,195,85,0,170,0,
     85,0,170,0: REM A="A",B="B"
```

```
1100 INK 1: FOR X=11 TO 21: FOR Y=0 TO 31:
     PRINT AT X,Y:"A": NEXT Y: NEXT X
1110 FOR X=0 TO 2: FOR Y=0 TO 31: PRINT AT
     X,Y;"B": NEXT Y: NEXT X
```

Una vez escritas estas líneas, introducir RUN 1000. Tras una pequeña pausa para que el ordenador recorra los gráficos, saldrá en la pantalla agua y cielo. Obsérvese que aunque se ha utilizado la misma INK 1 (azul) en los dos gráficos, el cielo es un poco más claro que el agua y ello es debido a la disposición de los números en la instrucción DATA de la línea 1020: el primer 8 es el agua y el segundo 8 es el cielo. A título de ejercicio, añadir un “0” (seguido por una coma) delante del primer 85 y borrar el “0” final; este nuevo programa traerá un cielo rosa. Una vez visto este efecto, cambiar los números del segundo lote de ochos para dejarlos como estaban y conseguir que el cielo siga siendo azul.

Como triángulo que haga las funciones de pirámide, optamos por el de la línea 210 de la primera parte del programa. Este triángulo será adecuado tras diversas modificaciones de su tamaño y su posición en la pantalla. Dicho de otra forma, la única modificación a introducir en la línea 210 para adaptarla a nuestras necesidades consiste en cambiar algunos números.

```
1120 FOR X=30 TO 0 STEP -1: PLOT 205-X,X-16
     2: DRAW X*2,0: NEXT X: REM PIRAMIDE
1130 FOR X=14 TO 0 STEP -1: PLOT 169-X,X-16
     0: DRAW X*2,0: NEXT X: REM PIRAMIDE
1140 FOR X=8 TO 0 STEP -1: PLOT 230-X,X-160
     : DRAW X*2,0: NEXT X: REM PIRAMIDE
```

Se han elegido las posiciones de estas pirámides dibujándolas en un tablero de melamina y haciendo pruebas después con diferentes tamaños. El cambio de STEP –2 por STEP –1 hace que los triángulos sean llenos, no rayados. El color sigue siendo azul porque al principio de la línea 110 había una INK 1;

para cambiarlo por otro color es necesario insertar otra orden INK.

El siguiente paso consiste en dibujar el barco y la vela. Vaya primero el casco del barco. Se puede hacer con tres secciones: una parte central y dos picos a proa y popa. Los triángulos 450 y 470 harán perfectamente la función de picos de proa y popa y para la parte central será adecuada la línea 230 si se hace la figura más larga y mucho más plana. Tras unas cuantas pruebas se llegó a este listado:

```
1150 FOR X=0 TO 20: PLOT 65+X,X-64: DRAW X*
     2,0: NEXT X: REM CASCO DE BARCO
1160 FOR X=20 TO 0 STEP -2: PLOT 187-X,X-64
     : DRAW -X*2,0: NEXT X: REM CASCO DE BA
     RCO
1170 FOR X=0 TO 10: PLOT 123-X*2.5,X+43: DR
     AW X*6,0: NEXT X: REM CASCO DE BARCO
```

La nota REM del final de cada línea sirve de ayuda a la hora de buscar en el listado la línea o líneas que producen el efecto que se quiere localizar. Estas líneas dibujan un casco de barco de razonable realismo. Es el momento de diseñar la vela y el mástil. El triángulo 790 parece el más prometedor, por lo que lo aprovecharemos con algunas modificaciones.

```
1180 FOR X=0 TO 50: PLOT X+104,55: DRAW -X*
     2,0+X*2: NEXT X: REM VELA DE BARCO
1190 PLOT 97,54: DRAW 20,60: PLOT 98,54: DR
     AW 20,60
```

Una vez escritas estas líneas, introducir RUN 1000 y saldrá en la pantalla una escena egipcia, bastante real si tenemos en cuenta que es el resultado de sólo once triángulos y algo de agua y cielo. Pero no tiene vida. Un camello o dos la llenarán de vida.

Ello obligará a introducir más caracteres gráficos, pero eso es fácil en este programa porque basta introducir un número

diferente en la primera parte de la línea 1010 FOR X=0 TO ? (el número de gráficos necesarios) y añadir luego algunas líneas DATA. No olvidar que el 0 cuenta como un número.

Para dibujar un camello y un hombre (aproximadamente a la misma escala) hay que generar seis caracteres gráficos más: cuatro para el camello (en una composición cuadrada) y dos para el hombre (uno encima de otro). Utilizando una hoja de papel (tal como se explicó en el capítulo 1), no resultó difícil y se hizo en seguida. Faltaba ubicarlos en la pantalla. Lo hace la línea 1200, que contiene las instrucciones encargadas de imprimir cuatro camellos y dos hombres.

Obsérvese que como el camello está constituido por cuatro caracteres y el hombre por dos, hay que tener mucho cuidado a la hora de situarlos para que las partes superiores coincidan con las inferiores. Se verá que la línea 1200 contiene diez instrucciones PRINT, pero sólo la primera contiene realmente la palabra PRINT. Todas las modificaciones hechas se muestran en el listado siguiente. Modificar la línea 1010 (0 a 7) y añadir luego las líneas 1030, 1040, 1050 y 1200. Irán a los respectivos huecos del programa pues se ha dejado sitio para ellas. Las letras gráficas de la línea 1200 están encerradas entre comillas para mostrar su carácter gráfico.

```
1010 FOR X=0 TO 7: FOR Y=0 TO 7: READ Z: PO
     KE USR CHR$ (144+X)+Y,Z: NEXT Y:NEXT X
1030 DATA 0,0,48,113,243,63,63,31,0,96,240,
     248,254,255,255,255:REM C="C",D="D"
1040 DATA 7,5,5,9,9,27,0,0,14,10,10,18,18,5
     4,0,0: REM E="E",F="F"
1050 DATA 0,0,0,24,24,8,28,60,60,60,124,124
     ,124,40,0,0: REM G="G",H="H"

1200 PRINT INK 2;AT 8,2;"GCD".AT 9,2;"HEF";
     AT 8,7;"CD";AT 9,7;"EF";AT 8,15;"CD";A
     T 9,15;"EF";AT8,18;"CD";AT 9,18;"EF";A
     T 8,21;"G";AT 9,21;"H"
1500 STOP
```

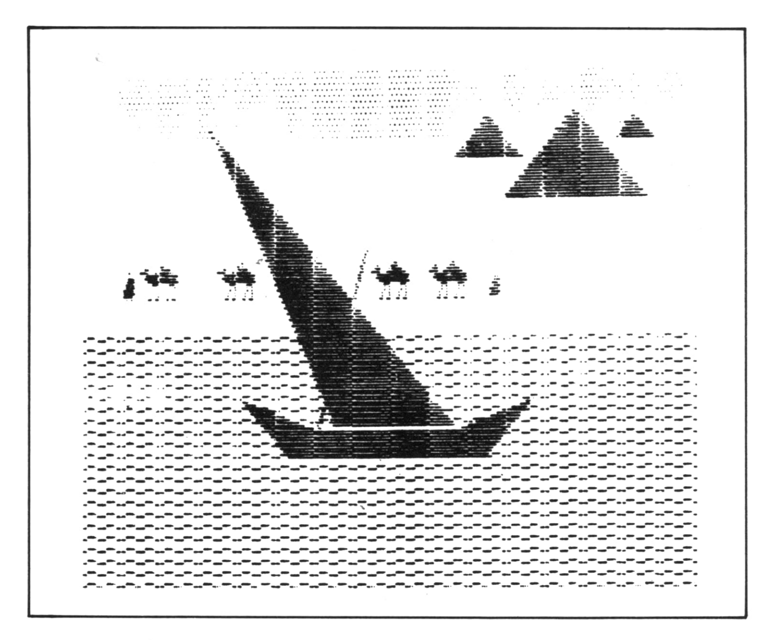


Figura 9.1. Escena egipcia desarrollada en este capítulo

Al ejecutar todo el programa una vez introducido este listado, la segunda parte del programa hace aparecer una fila de camellos y hombres en la otra ribera del Nilo. Son del color 2 para mayor contraste. La introducción de RUN con la escena del Nilo en la pantalla envía el programa de vuelta a los triángulos.

Este programa de referencia está ya completo y listo para su uso. Para aprovechar un triángulo cualquiera, introducir LIST y después el número impreso debajo del triángulo. Saldrá en la pantalla, señalada por un cursor, la línea que define ese triángulo. Copiar esa línea en papel; a partir de ahí será fácil plasmar la idea que se tenga en mente.

10. UN JUEGO TRIDIMENSIONAL

Para este capítulo final hemos elegido un juego de ceros y cruces, pero no un juego normal sino uno con tres tableros superpuestos. De esta forma, además de haber ocho "3-en-línea" ganadoras en un solo tablero, hay otras ocho líneas ganadoras en los otros dos tableros, dando un total de 24; y además se pueden hacer líneas ganadoras entre los tres tableros, perpendicularmente o diagonalmente. El resultado es un juego que tiene 49 posibles "3-en-línea" y que a medida que se va complicando la partida plantea más dificultades para planear movimientos con antelación.

El juego está diseñado para dos jugadores (el ordenador no participa). Lo primero que hay que hacer por lo tanto es diseñar la pantalla donde se jugará. Después de numerosos intentos de dibujar tres tableros que den una sensación de profundidad tridimensional (3-D), nos decidimos por tres tableros circulares superpuestos, pero ligeramente inclinados y un poco más gruesos en las proximidades del borde periférico delantero.

Los óvalos constituyen la mejor solución para representar tableros circulares creando la ilusión de que son vistos ligeramente desde arriba y desde un lado. Es algo difícil de describir pero fácil de comprender una vez visto.

En cualquier caso, el diseño de estos tableros saca a colación una parte de las rutinas PLOT/DRAW que todavía no hemos explorado: el dibujo de círculos. Por lo tanto, antes de pasar al programa propiamente dicho veremos un pequeño programa de introducción a las líneas curvas. Este programa comienza en la línea 7000, lejos del programa objeto de este capítulo. Escribir este pequeño programa y cuando se haya acabado introducir RUN 7000.

```
7000 REM PRUEBA
7010 CIRCLE 36,140,30
7020 PLOT 80,118: DRAW 43,0,-0.8*PI: PLOT 8
     0,118: DRAW 43,0,0.8*PI
```

En la pantalla aparecerá un círculo, resultado de la línea 7010, y un óvalo, resultado de la línea 7020. Esta última línea está compuesta por dos órdenes: cada una dibuja medio óvalo. Ahora como antes, los dos números o expresiones numéricas puestos a continuación de DRAW determinan el recorrido de la línea: el primer número gobierna la horizontalidad y el segundo (tras la coma) la verticalidad. En este caso sin embargo, esos números están modificados por un tercer número puesto también tras una coma (el que lleva la PI). Ese tercer número dice al ordenador que trace una curva. Cuando delante de ese número hay un signo menos, la curva va hacia arriba y cuando no hay signo menos va hacia abajo. PI sin más traza un semicírculo, pero modificado por 0.8 traza un semicírculo de curvatura corregida en esa proporción. La adición de dos líneas más a este pequeño programa resalta estos puntos.

```
7030 PLOT 144,118: DRAW 43,0,-PI: PLOT 144,
     118: DRAW 43,0,1.5*PI
7040 PLOT 24,32: DRAW 43,43,-0.9*PI: PLOT 2
     4,32: DRAW 43,43,0.9*PI
```

La primera parte de la línea 7030 dibuja un semicírculo por encima de la línea base y la segunda dibuja una parte y

media de un círculo por debajo de la línea base. La línea 7040 es la versión oval de la línea anterior, la que utilizaremos en nuestro juego para hacer el esquema de los tableros.

Grabar el programa 7000 con objeto de que se le pueda volver a llamar. De todas formas, conviene explicar los números de esas líneas. Los tres primeros números de la línea 7030 son: número de la columna en pixels, número de la fila en pixels y radio del círculo.

Los dos números PLOT de la línea 7020 son las coordenadas PIXEL (igual que el primero y segundo de la línea 7010). Los números DRAW son: el número de pixels correspondiente a la distancia horizontal de la línea a trazar, el número de pixels de la distancia vertical, y el grado y dirección de la curva. En la línea 7040 se ve que los dos números DRAW correspondientes a esas distancias vertical y horizontal son distintos de cero, luego la línea base está ligeramente inclinada hacia arriba de izquierda a derecha. Antes de seguir adelante, hacer algunas prácticas añadiendo más líneas a continuación de la 7040. Merece la pena estudiar estas órdenes sencillas hasta comprender perfectamente su acción.

El esqueleto o plan del juego de los ceros y las cruces será bastante sencillo. Estará constituido por una sección de generación de los caracteres gráficos (una cruz y un círculo densos y algunos números sesgados —1 al 9— para los tableros); la pantalla del juego se puede meter también en esta sección. Tendrá otra sección para los movimientos de las cruces, otra para los movimientos de los ceros y una sección pequeña para los marcadores. Esta última es necesaria porque con 49 posibles líneas ganadoras no es fácil recordar el marcador.

```
  10 REM * CEROS Y CRUCES TRIDIMENSIONAL *
  20 POKE 23609,255: POKE 23658,8
  30 BORDER 3: PAPER 6: CLS
  40 GOSUB 5000
1000 REM MOVIMIENTOS DE LAS X
```

```
1999 STOP
2000 REM MOVIMIENTOS DE LOS 0
2999 STOP
4000 REM MARCADOR
4999 STOP
5000 REM DISEÑO DE LA PANTALLA
5300 RETURN
5999 STOP
```

Una vez introducido este esqueleto en el ordenador se puede pasar a desarrollar los caracteres gráficos de la sección 5000. La X y el 0 serán caracteres gráficos pues deben ser más gruesos y, al igual que los números 1 al 9, ligeramente sesgados para dar la misma impresión de perspectiva que los tableros. El pequeño punto de la esquina inferior izquierda de la X y el 0 está hecho intenciònadamente: sin esos puntos, los caracteres borrarían una pequeña sección de cada línea divisoria extendida entre los números.

```
5010 FOR X=0 TO 10: FOR Y=0 TO 7: READ Z: P
     OKE USR CHR$ (144+X)+Y,Z: NEXT Y: NEXT
     X
5020 DATA 0,24,24,126,126,24,24,128,60,66,1
     29,129,129,129,66,188: REM A="A", B="B
     "
5030 DATA 0,64,96,80,8,6,4,0,0,48,72,72,10,
     12,8,0,0,48,72,92,2,2,28,0: REM C="C",
      D="D",E="E"
5040 DATA 0,64,100,88,120,100,66,0,0,32,76,
     82,34,2,12,0,0,32,76,82,34,34,28,0: RE
     M F="F",G="G", H="H"
5050 DATA 0,24,36,68,4,4,2,0,0,48,76,82,34,
     34,28,0,0,56,68,68,58,2,4,0: REM I="I"
     , J="J", K="K"
```

Una vez introducidas estas cinco líneas, llega el momento de hacer los tableros. En principio dibujaremos sólo un tablero para comprobar que todo es correcto, y una vez corre-

gidos los posibles problemas tendremos sentado el modelo para los otros dos.

```
5060 PLOT 15,118: DRAW 43,43,-0.9*PI: PLOT
     15,118: DRAW 43,43,0.9*PI: PLOT 15,118
     : DRAW 43,43,0.91*PI: PLOT 15,118: DRA
     W 43,43,0.92*PI
5070 PLOT 15,118: DRAW 43,43,0.93*PI: PLOT
     15,118: DRAW 43,43,0.94*PI: PLOT 15,11
     8: DRAW 43,43,0.95*PI
5080 PLOT 15,118: DRAW 43,43,0.96*PI: PLOT
     15,118: DRAW 43,43,0.97*PI: PLOT 15,11
     8: DRAW 43,43,0.98*PI
5090 PRINT AT 8,1;"A";AT 4,2;"C";AT 3,3;"D"
     ;AT 2,4;"E";AT 5,3;"F";AT 4,4;"G";AT 3
     ,5;"H";AT 6,4;"I";AT 5,5;"J";AT 4,6;"K"
5100 PLOT 16,128: DRAW 32,32: PLOT 24,120:
     DRAW 32,32: PLOT 20,148: DRAW 24,-24:
     PLOT 28,156: DRAW 24,-24: PLOT 15,118:
     DRAW -5,0: PLOT 15,118: DRAW 0,-5: PLO
     T 58,161: DRAW 5,0: PLOT 58,161: DRAW
     0,5
```

Las líneas 5060 a 5100 dibujan perfectamente el tablero. No prestar atención a las dos pequeñas V que hay a ambos lados del tablero; más adelante se explicará su razón de ser. En cualquier caso, el borde anterior del tablero, más grueso, sí necesita una explicación. El óvalo propiamente dicho se ha dibujado con las dos primeras instrucciones PLOT y DRAW de la línea 5060, pero después, para engrosar el borde del tablero, se ha incrementado en 0,01 el tercer parámetro de las instrucciones DRAW de la tercera y cuarta partes de la línea 5060 (0,91*PI y 0,92*PI). Esto hace que la curva se abra un poco en la segunda y tercera impresión. Las líneas 5070 y 5080 continúan este proceso. Todo ello ralentiza considerablemente el dibujo del tablero, pero es efectivo. El borde engrosado no es perfecto debido al método utilizado para dibujar las curvas, pero de todas formas sirve.

La línea 5100 dibuja las divisiones entre los números (puestos en los tableros con la única finalidad de permitir nombrar todas las posiciones). Mucho cuidado al introducir la línea 5090 pues la letra A mayúscula que hay cerca del principio no se introduce en el modo gráfico y sí los números sesgados. Las teclas asociadas a esos números son: C, D, E, F, G, H, I, J y K, respectivamente. Las líneas divisorias y las letras contribuyen ahora a aumentar la ilusión de inclinación.

El resto de la sección 5000 se puede introducir en muy poco tiempo, pues está constituido por dos repeticiones de las líneas 5060 y 5100 pero con diferentes números correspondientes a las posiciones. Mucho cuidado con ellos.

```
5110 PLOT 63,70: DRAW 43,43,-0.9*PI: PLOT 6
     3,70: DRAW 43,43,0.9*PI: PLOT 63,70: D
     RAW 43,43,0.91*PI: PLOT 63,70: DRAW 43
     ,43,0.92*PI
5120 PLOT 63,70: DRAW 43,43,0.93*PI: PLOT 6
     3,70: DRAW 43,43,0.94*PI: PLOT 63,70:
     DRAW 43,43,0.95*PI
5130 PLOT 63,70: DRAW 43,43,0.96*PI: PLOT 6
     3,70: DRAW 43,43,0.97*PI: PLOT 63,70:
     DRAW 43,43,0.98*PI
5140 PRINT AT 14,7;"B";AT 10,8;"C";AT 9,9;"
     D";AT 8,10;"E";AT 11,9;"F";AT 10,10;"G
     ";AT 9,11;"H";AT 12,10;"I";AT 11,11;"J
     ";AT 10,1;"K"
5150 PLOT 64,80: DRAW 32,32: PLOT 72,72: DR
     AW 32,32: PLOT 68,100: DRAW 24,-24: PL
     OT 76,108: DRAW 24,-24: PLOT 63,70: DR
     AW -5,0: PLOT 63,70: DRAW 0,-5: PLOT 1
     06,113: DRAW 5,0: PLOT 106,113: DRAW 0
     ,5
5160 PLOT 111,22: DRAW 43,43,-0.9*PI: PLOT
     111,22: DRAW 43,43,0.9*PI: PLOT 111,22
     : DRAW 43,43,0.91*PI: PLOT 111,22: DRA
     W 43,43,0.92*PI
```

```
5170 PLOT 111,22: DRAW 43,43,0.93*PI: PLOT
     111,22: DRAW 43,43,0.94*PI: PLOT 111,2
     2: DRAW 43,43,0.95*PI
5180 PLOT 111,22: DRAW 43,43,0.96*PI: PLOT
     111,22: DRAW 43,43,0.97*PI: PLOT 111,2
     2: DRAW 43,43,0.98*PI
5190 PRINT AT 20,13;"C";AT 16,14;"C";AT 15,
     15;"D";AT 14,16;"E";AT 17,15;"F";AT 16
     ,16;"G";AT 15,17;"H";AT 18,16;"I";AT 1
     7,17;"J";AT 16,18;"K"
5200 PLOT 120,24: DRAW 32,32: PLOT 112,32:
     DRAW 32,32: PLOT 116,52: DRAW 24,-24:
     PLOT 124,60: DRAW 24,-24: PLOT 111,22:
     DRAW -5,0: PLOT 111,22: DRAW 0,-5: PLO
     T 154,65: DRAW 5,0: PLOT 154,65: DRAW
     0,5
5210 PLOT 60,160: DRAW 96,-96: PLOT 15,118:
     DRAW 96,-96
```

Estas líneas completan el dibujo de los tres tableros. La línea 5210 dibuja dos líneas rectas que unen los tres tableros. Las peculiares marcas en V mencionadas anteriormente pueden descubrir ahora su propósito: son “tuercas de mariposa” puestas ahí para atornillar los tres tableros al marco. Ponen también su granito de arena para crear la ilusión de las tres dimensiones.

Queda por añadir un título junto con algunas instrucciones del juego y el marcador.

```
5220 PRINT INVERSE 1; INK 1;AT 15,1;"CEROS"
     ;AT 17,1;" Y CRUCES";AT 19,1;"EN 3-D"
5230 PRINT AT 1,11;"HAY 49 FORMAS DE";AT 2,
     12;"UNIR 3 EN UNA LINEA";AT 3,13;"----
     ---------------";AT 4,14;"X SALE"
5240 PRINT AT 5,15"----------------";AT 6,1
     6;"ESCRIBIR LETRA";AT 7,17;"Y LUEGO NU
     MERO"
```

```
5250 PRINT AT 8,18;"------------";AT 9,19;
     "GANA QUIEN";AT 10,20;"TIENE MAS 3";AT
     11,21;"EN LINEA";AT 12,22;"---------"
5260 PRINT AT 15,22;"PUNTOS X:";AT 17,22;"P
     UNTOS 0:"
5270 LET VX=0: LET V0=0
5280 PRINT AT 15,30;VX;AT 17,30;V0
```

Estas líneas completan la sección 5000. Obsérvese que se ha asignado la variable VX al marcador de las X y la variable V0 al marcador de los 0. Ambas se utilizarán en la sección 4000, la encargada de llevar el marcador. La línea 5300 RETURN envía el ordenador a la línea 1000 (la línea siguiente a la GO SUB de origen), y allí iremos también nosotros para comenzar a desarrollar las líneas que rigen los movimientos de las X. Pero antes será mejor ejecutar el programa tal como está para ver cómo es la pantalla. Si algo va mal, repasar todas las entradas relacionadas con ese fallo. Si se ha seguido esta descripción paso a paso se estará preparado para localizar rápidamente cualquier fallo.

La sección 1000 del programa (movimientos de las X) ha de tener una INPUT que pregunte dónde tiene que poner X, y tras ella debe ir una larga serie de órdenes IF/THEN que digan al ordenador lo que tiene que hacer en las 27 coordenadas posibles que se le puede haber dado. La última letra mayúscula de las líneas 1050 a 1130 es "A" en el modo gráfico.

```
1010 INPUT "X MUEVE A (LETRA):";U$
1020 INPUT "X MUEVE A (NUMERO):";V$
1050 IF U$="A" AND V$="1" THEN PRINT AT 4,2
     ;"A"
1060 IF U$="A" AND V$="2" THEN PRINT AT 3,3
     ;"A"
1070 IF U$="A" AND V$="3" THEN PRINT AT 2,4
     ;"A"
1080 IF U$="A" AND V$="4" THEN PRINT AT 5,3
     ;"A"
```

```
1090 IF U$="A" AND V$="5" THEN PRINT AT 4,4
     ;"A"
1100 IF U$="A" AND V$="6" THEN PRINT AT 3,5
     ;"A"
1110 IF U$="A" AND V$="7" THEN PRINT AT 6,4
     ;"A"
1120 IF U$="A" AND V$="8" THEN PRINT AT 5,5
     ;"A"
1130 IF U$="A" AND V$="9" THEN PRINT AT 4,6
     ;"A"
```

Una vez introducidas las dos preguntas INPUT y los nueve números del tablero A, se puede pasar a probar el juego. Responder a la pregunta INPUT con la letra A (pues se trata del tablero A) y después un número. Cuando salga al pie de la pantalla el signo STOP, escribir GO TO 1000 y pulsar ENTER para que salga otra vez la INPUT. Se pueden comprobar así todas las entradas A. Si sale una X en un lugar equivocado, es debido a que no se ha contestado bien a la INPUT o a que se ha cometido un error al introducir el listado.

Si esas líneas no plantean ningún problema se puede pasar a introducir las correspondientes a los tableros B y C, añadiendo una línea al final para dirigir el ordenador a la sección del marcador. Las "A" del final de las líneas 1200 a 1480 son gráficas.

```
1200 IF U$="B" AND V$="1" THEN PRINT AT 10,
     8;"A"
1210 IF U$="B" AND V$="2" THEN PRINT AT 9,9
     ;"A"
1220 IF U$="B" AND V$="3" THEN PRINT AT 8,1
     0;"A"
1230 IF U$="B" AND V$="4" THEN PRINT AT 11,
     9;"A"
1240 IF U$="B" AND V$="5" THEN PRINT AT 10,
     10;"A"
```

```
1250 IF U$="B" AND V$="6" THEN PRINT AT 9,1
     1;"A"
1260 IF U$="B" AND V$="7" THEN PRINT AT 12,
     10;"A"
1270 IF U$="B" AND V$="8" THEN PRINT AT 11,
     11;"A"
1280 IF U$="B" AND V$="9" THEN PRINT AT 10,
     12;"A"
1400 IF U$="C" AND V$="1" THEN PRINT AT 16,
     14;"A"
1410 IF U$="C" AND V$="2" THEN PRINT AT 15,
     15;"A"
1420 IF U$="C" AND V$="3" THEN PRINT AT 14,
     16;"A"
1430 IF U$="C" AND V$="4" THEN PRINT AT 17,
     15;"A"
1440 IF U$="C" AND V$="5" THEN PRINT AT 16,
     16;"A"
1450 IF U$="C" AND V$="6" THEN PRINT AT 15,
     17;"A"
1460 IF U$="C" AND V$="7" THEN PRINT AT 18,
     16;"A"
1470 IF U$="C" AND V$="8" THEN PRINT AT 17,
     17;"A"
1480 IF U$="C" AND V$="9" THEN PRINT AT 16,
     18;"A"
1500 GO SUB 4000
```

Las líneas 1200 a 1500 completan los movimientos de las X. La línea 1500 envía el ordenador a la 4000, y ahí vamos nosotros también. Cada movimiento de las X o de los 0 irá seguido por una señal INPUT que pregunta “Hay un 3-en-línea?”. Para evitar pasarla por alto durante el desarrollo del juego, será una INPUT intermitente. La respuesta normal será N (NO), en cuyo caso saldrá la INPUT correspondiente al siguiente movimiento. Si la respuesta es S (SI), es incrementada en 1 la casilla de los puntos de ese jugador, suena

una serie de pitidos y sale a escena la INPUT del siguiente movimiento. Esta sección 4000 ocupa sólo seis líneas.

```
4010 INPUT FLASH 1;"HAY 3-EN-LINEA? (S/N)";
     H$
4020 IF H$="S" THEN LET VX=VX+1: PRINT AT 1
     5,30;VX: FOR X=0 TO 20: BEEP .1,X: NEX
     T X: GO TO 2000
4030 IF H$="N" THEN RETURN
4510 INPUT FLASH 1;"HAY 3-EN-LINEA? (S/N)";
     J$
4520 IF J$="S" THEN LET VO=VO+1: PRINT AT 1
     7,30;VO: FOR X=0 TO 20: BEEP .1,X: NEX
     T X: GO TO 1000
4530 IF J$="N" THEN RETURN
```

Se puede poner a prueba esta sección respondiendo "S" a la pregunta "Hay 3-en-línea?" con independencia de que lo haya o no. Si se introduce una S, el programa incrementa en 1 esa casilla del marcador y se oyen los pitidos antes de que salga el signo STOP. La primera parte de la sección 4000 envía el programa a la sección 2000, que no ha sido introducida todavía. Lo haremos ahora. La "B" del final de las líneas es gráfica.

```
2010 INPUT "O MUEVE A (LETRA):";X$
2020 INPUT "O MUEVE A (NUMERO):";Y$
2050 IF X$="A" AND Y$="1" THEN PRINT AT 4,2
     ;"B"
2060 IF X$="A" AND Y$="2" THEN PRINT AT 3,3
     ;"B"
2070 IF X$="A" AND Y$="3" THEN PRINT AT 2,4
     ;"B"
2080 IF X$="A" AND Y$="4" THEN PRINT AT 5,3
     ;"B"
2090 IF X$="A" AND Y$="5" THEN PRINT AT 4,4
     ;"B"
```

```
2100 IF X$="A" AND Y$="6" THEN PRINT AT 3,5
     ;"B"
2110 IF X$="A" AND Y$="7" THEN PRINT AT 6,4
     ;"B"
2120 IF X$="A" AND Y$="8" THEN PRINT AT 5,5
     ;"B"
2130 IF X$="A" AND Y$="9" THEN PRINT AT 4,6
     ;"B"
2200 IF X$="B" AND Y$="1" THEN PRINT AT 10,
     8;"B"
2210 IF X$="B" AND Y$="2" THEN PRINT AT 9,9
     ;"B"
2220 IF X$="B" AND Y$="3" THEN PRINT AT 8,1
     0;"B"
2230 IF X$="B" AND Y$="4" THEN PRINT AT 11,
     9;"B"
2240 IF X$="B" AND Y$="5" THEN PRINT AT 10,
     10;"B"
2250 IF X$="B" AND Y$="6" THEN PRINT AT 9,1
     1;"B"
2260 IF X$="B" AND Y$="7" THEN PRINT AT 12,
     10;"B"
2270 IF X$="B" AND Y$="8" THEN PRINT AT 11,
     11;"B"
2280 IF X$="B" AND Y$="9" THEN PRINT AT 10,
     12;"B"
2400 IF X$="C" AND Y$="1" THEN PRINT AT 16,
     14;"B"
2410 IF X$="C" AND Y$="2" THEN PRINT AT 15,
     15;"B"
2420 IF X$="C" AND Y$="3" THEN PRINT AT 14,
     16;"B"
2430 IF X$="C" AND Y$="4" THEN PRINT AT 17,
     15;"B"
2440 IF X$="C" AND Y$="5" THEN PRINT AT 16,
     16;"B"
2450 IF X$="C" AND Y$="6" THEN PRINT AT 15,
     17;"B"
```

```
2460 IF X$="C" AND Y$="7" THEN PRINT AT 18,
     16;"B"
2470 IF X$="C" AND Y$="8" THEN PRINT AT 17,
     17;"B"
2480 IF X$="C" AND Y$="9" THEN PRINT AT 16,
     18;"B"
2500 GO SUB 4000
```

Son las mismas series de entradas que había en la sección 1000 pero ahora con las variables X$ e Y$ en vez de U$ y V$, y con la "B" gráfica, el 0. El método seguido aquí para introducir tan tedioso listado, válido siempre que haya que repetir (más o menos) líneas, consiste en introducir la primera línea que sigue a la INPUT (2050), llamarla al pie de la pantalla para EDITarla, modificar su número de línea cambiando el 1 por el 2, cambiar las coordenadas y luego pulsar ENTER. Así lo hicimos con las nueve primeras líneas y luego cambiamos además la A por la B y seguimos con las nueve líneas siguientes. De esta forma hay menos oportunidades de cometer errores.

Una vez introducido este listado llega el momento de jugar una partida para probar todo el programa. Las complejidades de las diversas combinaciones 3-en-línea posibles empezarán a surgir pronto. El juego es fascinante incluso jugando contra uno mismo y exige reflexionar mucho antes de hacer un movimiento pues son muy numerosas las alternativas de formar un 3-en-línea. La figura 10.1 muestra la pantalla del juego.

Sólo falta limpiar el programa quitando todos los signos STOP (1999, 2999, 4999), que ya no hacen nada útil, y grabar el programa en cinta (con SAVE). Este juego proporcionará diversión durante mucho tiempo. Cuanta más práctica se consiga jugando contra uno mismo, más preparado se estará para batir a otros oponentes, porque se necesita mucha destreza.

Quien haya introducido los juegos uno por uno, comenzando por el capítulo 1, habrá adquirido cierto dominio de los

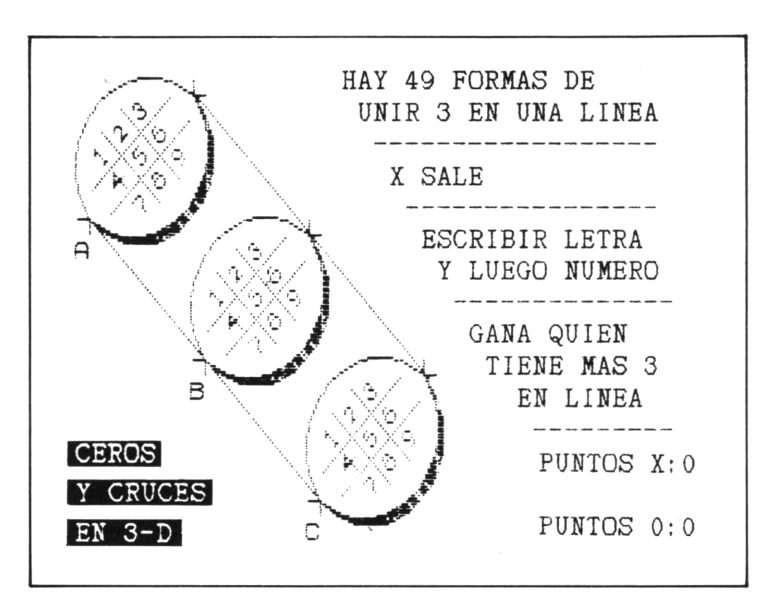


Figura 10.1

aspectos más sencillos del BASIC. Si se sigue haciendo siempre un esqueleto o plan antes de comenzar a escribir un programa, se estará en el buen camino.

ORDENADOR PERSONAL

Títulos publicados

- **Cómo funcionan los ordenadores**
- **Cómo programar con ZX-Spectrum**
- **Práctica del Amstrad PCW8256/512**
- **Secretos del Commodore 64**
- **Diseño de juegos con Amstrad**
- **Cómo hacer programas que funcionen**
- **Introducción al Basic MSX**
- **Programas de juegos con el ZX Spectrum y Spectrum +**

ELECTRONICA BASICA

Títulos publicados

- **Introducción a la electrónica**
- **Interpretación de esquemas**
- **Elementos de Electrónica 1**
  (Componentes y circuitos básicos)
- **Elementos de Electrónica 2**
  (Teoría de la corriente alterna)
- **Elementos de Electrónica 3**
  (Tecnología de los semiconductores)
- **Elementos de Electrónica 4**
  (Sistemas y circuitos de microprocesadores)
- **Elementos de Electrónica 5**
  (Telecomunicaciones)
- **Elementos de Electrónica 6**
  (Audio)

CIRCUITOS PRACTICOS DE ELECTRONICA
Títulos publicados

- **55 circuitos especiales de baja frecuencia**
- **71 circuitos con transistores**
- **Triacs y tiristores**
- **Transistores MOS**
- **Circuitos Impresos**
- **Juegos electrónicos en TV**
- **Fuentes de alimentación electrónica**

- Este libro se ha elaborado como una guía paso a paso y sirve de ayuda para escribir programas de juegos gráficos en los populares ZX Spectrum y Spectrum+ de Sinclair. Los juegos comienzan desde el nivel más sencillo y van progresando hasta llegar a un juego tridimensional.
- Si se estudian los programas en el orden mostrado en el libro, en muy poco tiempo se conocerá mucho mejor el BASIC Sinclair y la forma de planificar lógicamente los programas.
- Este libro es una lectura obligatoria para todos los usuarios del ZX Spectrum y del Spectrum+ , y en particular para los principiantes en las técnicas de programación.

